

Novel Methods for the Analysis of Small Molecule Fragmentation Mass Spectra

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik

der Friedrich-Schiller-Universität Jena

von Dipl.-Bioinf. Franziska Hufsky

geboren am 25. Nov. 1986 in Zeulenroda, Deutschland

Gutachter:

1. Prof. Dr. Sebastian Böcker, Friedrich-Schiller-Universität Jena
2. Prof. Dr. Georg Pohnert, Friedrich-Schiller-Universität Jena
3. Prof. Dr. Markus Chimani, Universität Osnabrück

Tag der öffentlichen Verteidigung: 14. März 2014

Gedruckt auf alterungsbeständigem Papier nach DIN-ISO 9706

Abstract

Small molecules that arise as intermediates and products during all life-sustaining chemical reactions within the cells of living organisms are called metabolites. The identification of these small molecules in a high throughput manner plays an important role for the development of new drugs, the search for biomarkers, the identification of drug degradation products, the elucidation of metabolic networks of organisms, and for many further research areas.

Nuclear magnetic resonance enables full structural elucidation of unknown molecules, but requires large amounts of sample material. In contrast, mass spectrometry (MS) is much more sensitive and can be used in high-throughput experiments. Additional fragmentation of the molecules is used to obtain information beyond the mass of the molecule. The identification of unknown metabolites is still the major bottleneck in the analysis of (fragmentation) mass spectra. A significant number of metabolites are still unknown, and therefore not listed in any database. For these molecules, conventional methods based on spectral comparison or known molecular structures cannot be applied. In this work, we present computational methods for the analysis of fragmentation mass spectra of unknown small molecules that cannot be found in any database.

Gas chromatography-mass spectrometry is one of the oldest and most widespread techniques for the analysis of small molecules. The common ion source for this analytical setup is electron ionization (EI), which simultaneously leads to fragmentation of the molecule. For unstable molecular ions this may lead to a full fragmentation; thus, the molecular ion peak is often barely visible in the mass spectrum or even absent. The fragmentation of small molecules by electron ionization is already well understood; the manual interpretation of the fragmentation mass spectra, however, is cumbersome, time consuming and requires expert knowledge. Automated methods for the analysis of EI mass spectra are currently confined to database search and rule-based approaches. We present a method for reconstructing fragmentation patterns of small molecules from high mass accuracy EI spectra. The calculated fragmentation trees annotate the peaks in the mass spectrum with molecular formulas of fragments and explain relevant fragmentation pathways. Fragmentation trees enable the identification of the molecular ion and its molecular formula if the molecular ion is present in the spectrum. The method works even if the molecular ion is of very low abundance. Mass spectrometry experts confirm that the calculated trees correspond very well to known fragmentation mechanisms.

Further, fragmentation trees are used for classification and identification of unknown metabolites. By comparing trees, structural and chemical similarities to already-known molecules can be determined. We use pairwise local alignments of fragmentation trees for this task. In order to compare a fragmentation tree of an unknown metabolite to a huge database of fragmentation trees, fast algorithms for solving the tree alignment problem are required. Unfortunately the alignment of unordered trees, such as fragmentation trees, is NP-hard. We present three exact algorithms for the problem: a dynamic programming (DP) algorithm, a sparse variant of the DP algorithm, and an integer linear programming (ILP) algorithm. Somewhat unexpectedly, the ILP is clearly outperformed by both DP

approaches. Evaluation of our methods showed that thousands of alignments can be computed in a matter of minutes using DP.

Both the computation and the comparison of fragmentation trees are rule-free approaches that require no chemical knowledge about the unknown molecule. Thus, the presented methods will be very helpful in the automated analysis of metabolites that are not included in common libraries, and have the potential to support the explorative character of metabolomics studies.

Zusammenfassung

Kleine Moleküle, die als Zwischen- und Endprodukte aller lebenserhaltenden chemischen Reaktionen innerhalb der Zellen lebender Organismen entstehen, werden als Metaboliten bezeichnet. Die Identifizierung dieser kleinen Moleküle im Hochdurchsatzverfahren spielt eine große Rolle für die Entwicklung neuer Medikamente, die Suche nach Biomarkern, die Identifizierung von Drogenabbauprodukten, die Aufklärung metabolischer Netzwerke von Lebewesen sowie für viele weitere Forschungsgebiete.

Kernspinresonanzspektroskopie ermöglicht die vollständige Strukturaufklärung unbekannter Moleküle, jedoch werden für dieses Verfahren große Mengen an Probensubstanz benötigt. Massenspektrometrie (MS) ist hingegen wesentlich sensitiver und kann daher auch im Hochdurchsatz angewandt werden. Durch zusätzliche Fragmentierung der Moleküle gewinnt man Informationen über die Molekülmasse hinaus. Die Identifizierung unbekannter Metaboliten ist noch immer die Hauptproblematik bei der Analyse von (Fragmentierungs-)Massenspektren. Eine erhebliche Zahl von Metaboliten ist bis heute unbekannt, und somit in keiner Datenbank gelistet. Konventionelle Methoden, basierend auf Spektrenvergleich oder bereits bekannten Strukturformeln, stoßen daher häufig an ihre Grenzen. In dieser Arbeit stellen wir Computer-gestützte Methoden zur Analyse von Fragmentierungsmassenspektren unbekannter kleiner Moleküle, die in keiner Datenbank zu finden sind, vor.

Gaschromatographie mit Massenspektrometrie-Kopplung ist eine der ältesten und wichtigsten Verfahren zur Analyse kleiner Moleküle. Die dabei am häufigsten verwendete Ionenquelle ist die Elektronenstoßionisation (EI), welche zusätzlich zur Fragmentierung des Moleküls führt. Bei instabilen Molekülionen kann es bis zu einer vollständigen Fragmentierung kommen, sodass der Peak des Molekülions im Massenspektrum kaum sichtbar ist oder gar fehlt. Die Fragmentierung kleiner Moleküle durch Elektronenstoßionisation ist bereits gut verstanden; die manuelle Interpretation dieser Fragmentierungsmassenspektren ist jedoch umständlich, zeitaufwendig und erfordert Expertenwissen. Automatische Methoden für die Analyse von EI-Massenspektren beschränken sich derzeit auf Datenbanksuche und regelbasierte Ansätze. Wir präsentieren eine Methode zur Rekonstruktion von Fragmentierungsmustern von kleinen Molekülen aus hochaufgelösten EI-Massenspektren mit hoher Massengenauigkeit. Die dabei berechneten Fragmentierungsbäume annotieren die Peaks im Massenspektrum mit Molekülformeln von Fragmenten und deuten auf relevante Fragmentierungswege hin. Mittels dieser Fragmentierungsbäume lässt sich das Molekülion und dessen Molekülformel bestimmen. Die Methode funktioniert sogar für sehr komplexe Spektren, sowie Spektren, in denen das Molekülion kaum sichtbar ist. Massenspektrometrie-Experten bestätigen, dass die berechneten Bäume bekannte Fragmentierungsmechanismen widerspiegeln.

Im nächsten Schritt können die Fragmentierungsbäume zur Klassifizierung und Identifizierung von unbekannten Molekülen verwendet werden. Mittels Baumvergleich lassen sich strukturelle und chemische Ähnlichkeiten zu anderen, bereits bekannten Molekülen feststellen. Hierfür werden paarweise lokale Alignments von Fragmentierungsbäumen verwendet. Um den Fragmentierungsbaum eines unbekannten Metaboliten mit einer großen

Fragmentierungsbaumdatenbank vergleichen zu können, sind schnelle Algorithmen zur Lösung des Baumalignmentproblems erforderlich. Leider ist das Alignieren von ungeordneten Bäumen, zu denen die Fragmentierungsbäume zählen, NP-schwer. Wir stellen drei exakte Algorithmen zur Berechnung von Baumalignments vor: Zwei dieser Algorithmen basieren auf dynamischer Programmierung (DP), wobei der zweite Algorithmus auf dem ersten aufbaut, indem er die dünne Besetzung der DP Tabelle zu seinen Gunsten nutzt. Der dritte Algorithmus basiert auf ganzzahliger linearer Programmierung (ILP). Wider Erwarten sind die Laufzeiten der beiden DP Algorithmen wesentlich geringer als die des ILP Algorithmus. Tausende Alignments können mittels DP in wenigen Minuten berechnet werden.

Sowohl die Berechnung als auch der Vergleich von Fragmentierungsbäumen sind regelfreie Ansätze, die keine chemischen Kenntnisse über das zu untersuchende Molekül voraussetzen und können daher für die automatische Analyse von Fragmentierungsmassenspektren unbekannter Metaboliten verwendet werden.

Acknowledgements

Completing my PhD thesis has been probably the most ambitious activity of my life so far. One of the joys of completion is to look over the journey past and remember all the friends and family who shared the best moments with me and helped me to get through the difficult times.

First of all, I would like to thank my supervisor Sebastian Böcker. He has given me the freedom and confidence to go my own way, and supported me with lots of valuable ideas. His door was always open for questions and insightful discussions about various problems.

I gratefully acknowledge the funding and training program received from the International Max Planck Research School fellowship. I am also grateful to the funding received through the University of Jena and the Deutsche Forschungsgemeinschaft (DFG), within the project “Identifying the Unknowns” (BO 1910/10), to undertake my PhD.

I am grateful to all current and former members of our research group for an enjoyable and inspiring working atmosphere. Special thanks go to the “fragmentation tree team”, Florian Rasche, Kerstin Scheubert and Kai Dührkop, for many fruitful discussions and suggestions and being great co-authors in most of my publications. In particular, I am very grateful to Florian Rasche for his scientific advice and knowledge while learning the roots of fragmentation trees. It’s been a pleasure working with you! I highly appreciate my office neighbors Thasso Griebel and Kerstin Scheubert and want to thank them for being both smart and fun. A special thank goes to Kathrin Schowtka for her kindness and patience while helping me with bureaucracy and administrative work. Thanks for making our team run smoothly.

I wish to acknowledge all my collaborators for very successful cooperations. In particular, I want to thank Georg Pohnert, and Martin Rempt from the Institute for Inorganic and Analytical Chemistry, Bioorganic Analytics group (FSU Jena, Germany), and Aleš Svatoš from the Research Group Mass Spectrometry and Proteomics (MPI-CE Jena, Germany) for their explanations, their patience and their trust in our work. Further, I want to thank Markus Chimani from the Theoretical Computer Science group (Osnabrück University, Germany) for the fruitful cooperation during the algorithm development.

This thesis would not have been possible without all the people, that were supplying us with reference data: I want to thank Martin Rempt and Georg Pohnert (Bioorganic Analytics, FSU Jena, Germany) for measuring the EI spectra; Aleš Svatoš, Marco Kai, and Ravi Kumar Maddula (MPI-CE Jena, Germany) for measuring the Orbitrap data; Miroslav Strnad (Palacký University, Olomouc, Czech Republic) for supplying the zeatins, and Evangelos Tatsis (MPI-CE Jena, Germany) for poppy samples; Masanori Arita (University of Tokyo, Japan) for providing the MassBank data; and David Grant and Dennis Hill (University of Connecticut, Storrs, USA) for sharing their data.

Several students worked on their theses in the fragmentation tree team. I wish to acknowledge Kai Dührkop and Marcus Ludwig for their excellent work. I further want to thank the student assistants for implementing our algorithms and dealing with the software development.

I warmly appreciate Tim White, Kerstin Scheubert, Marcus Ludwig, Sascha Winter and Markus Fleischauer for proofreading parts of this thesis and giving advices to improve legibility.

I feel myself lucky to have lasting friends that supported me during disappointments and happiness in both my professional and my personal life. Thanks for being there for me. Finally, I would like to thank my mom and dad, my brother and my grandparents for their infinite support throughout everything. Thank you, for constantly supporting me on my way even if you are not always so clear about what I am actually doing. Danke für Alles. Es ist schön, dass es euch gibt!

Preface

This thesis covers large parts of my research in the automated analysis of fragmentation mass spectra of metabolites for the last four years. During this time, I was working at the Bioinformatics Group of Professor Sebastian Böcker at the Friedrich-Schiller Universität Jena. My research was financed by a scholarship from the International Max Planck Research School Jena and later by the university’s basic funding and the project “IDUN” funded by the Deutsche Forschungsgemeinschaft (DFG).

Most of the results presented in this work have been published [57, 60, 61, 113] and have been achieved in cooperation with my supervisor Sebastian Böcker, our collaborators Georg Pohnert, Martin Rempt, Markus Chimani, Aleš Svatoš and Marco Kai, my colleagues Florian Rasche, Kerstin Scheubert and Kai Dührkop and former diploma student Thomas Zichner.

I also participated in the calculation of fragmentation trees from MSⁿ data [124, 125], the computation of characteristic substructures for metabolite classes [93], and the estimation of the abundance of heavy isotopes in peptides by isotope pattern analysis [154]. Together with my supervisor Sebastian Böcker and my colleague Kerstin Scheubert I have written several review articles on the computational analysis of fragmentation mass spectra of metabolites [62, 63, 126]. Before starting my research in computational mass spectrometry, I have written my diploma thesis in the field of parameterized algorithms [58, 59].

This thesis consists of six chapters. The main results of this thesis are presented in Chapters 4 and 5.

Chapter 4 describes the calculation and evaluation of fragmentation trees from electron ionization mass spectra using results from both [61] and [57]. Sebastian Böcker and I developed the extended fragmentation tree concept. Adaption of the scoring function to EI data, algorithm implementation, and calculation of fragmentation trees was done by me. Chemical analysis of the results (see Section 4.2.4) was done by Martin Rempt [115]. Molecular formula evaluation, comparisons to other methods, and evaluation against annotated fragmentation pathways has been carried out by me.

Chapter 5 deals with local tree alignments for the automated comparison of fragmentation trees. The basic concept has been introduced by Rasche *et al.* [113]. In this project, I participated in the development of the scoring and carried out the compound clustering based on fragmentation tree similarity. The clustering approach (see Section 5.4) has been already presented by Florian Rasche in his thesis [111] and is reproduced here only for the sake of completeness to demonstrate a biological use-case. Three fast algorithms for the alignment problem have been presented by me at the *20th Annual International Conference on Intelligent Systems for Molecular Biology* (ISMB 2012) [60]. I developed the algorithms together with all co-authors and performed large parts of the evaluation.

For the remainder of this thesis, I will use “we” as the first person pronoun, as it is common in scientific literature. This may be interpreted as “the reader and I” or as “my collaborators and I”, whichever suits best in the situation.

Contents

1	Introduction	1
1.1	The Rise of Metabolomics in the “Omics-Era”	1
1.2	Contribution of this Work	3
2	Biological Background, Analytical Concepts and Theoretical Notation	5
2.1	Molecules	5
2.2	Metabolites	6
2.3	Mass Spectrometry	7
2.3.1	Gas Phase Ion Generation Techniques	8
2.3.2	Mass Analyzer	9
2.4	Common Fragmentation Mass Spectrometry Setups	10
2.4.1	Gas Chromatography Electron Ionization MS	10
2.4.2	Liquid Chromatography Collision-induced Dissociation MS	12
2.4.3	Further Experimental Setups	13
2.5	Graph-Theoretical Notation	13
3	Computational Analysis of Small Molecule MS Data	15
3.1	Reference Data	15
3.2	Compound Identification and Structure Elucidation	16
3.3	Searching in Spectral Libraries	16
3.4	Molecular Formula Identification	17
3.5	Searching in Molecular Structure Databases	18
3.5.1	Rule-based Fragmentation Spectrum Prediction	19
3.5.2	Combinatorial Fragmentation	20
3.5.3	Predicting Structural Features and Compound Classes	20
3.6	Fragmentation Trees	21
4	Fragmentation Trees for Electron Ionization Mass Spectra	25
4.1	Computing Fragmentation Trees	25
4.1.1	General Fragmentation Model	25
4.1.2	Weighting the Fragmentation Graph	26
4.1.3	Calculating Fragmentation Trees	27
4.1.4	Handling Hard Ionization Issues	30
4.1.5	Dealing with Derivatizations	32
4.2	Evaluation of Fragmentation Tree Quality	32
4.2.1	Datasets and Parameter Settings	33
4.2.2	Identification of Molecular Ion Peaks and Molecular Formulas	33
4.2.3	Evaluation against Annotated Fragmentation Pathways	36
4.2.4	Evaluation against Expert Knowledge	40
4.2.5	Evaluation against MetFrag	44

5	Fragmentation Tree Alignment	47
5.1	Formal Problem Definition	47
5.2	Alignment Algorithms	50
5.2.1	Dynamic Programming	50
5.2.2	Sparse Dynamic Programming	55
5.2.3	Integer Linear Programming	58
5.3	Comparing Running Times of the Algorithms	59
5.3.1	Reference Datasets	60
5.3.2	Running Time Comparison	60
5.4	Application of Fragmentation Tree Alignments: Clustering Similar Com- pounds	63
5.4.1	MS Datasets	63
5.4.2	Scoring Alignments	64
5.4.3	Normalization of Scores and Fingerprinting	65
5.4.4	Clustering Fragmentation Trees	66
5.4.5	Clustering Results of the Reference Dataset	67
5.4.6	Identifying Unknowns from a Biological Sample	67
6	Conclusion	71
6.1	Future Work	73
A	Appendix	91

1 Introduction

The computer is incredibly fast, accurate, and stupid.
Man is unbelievably slow, inaccurate, and brilliant.
The marriage of the two is a force beyond calculation.

Leo Cherne

1.1 The Rise of Metabolomics in the “Omics-Era”

Ever since the rise of genomics, the suffix ‘-omics’ has been added to the names of many fields to denote the collective characterization and quantification of pools of biological molecules on a large scale. *Genomics* studies events that can happen by determining the entire DNA sequence of organisms. Numerous genomes have been sequenced in the last two decades¹. But “by studying a brick one cannot learn anything about the design of a building nor the architect” [156]. So, nowadays focus is shifting from structural towards functional genomics, aiming to comprehend an organism’s response to a conditional perturbation: *Transcriptomics* studies mRNA expression levels capturing what appears to be happening and *proteomics* examines the events that actually are happening by large-scale study of proteins, particularly their structures and functions. But even mRNA gene expression data and proteomic analysis do not tell the whole story of what is happening in a cell. The ultimate endpoint measurement of biological events linking genotype to phenotype is *metabolomics* which can give an instantaneous snapshot of the physiology of a cell.

Metabolomics is the systematic study of the unique chemical fingerprints that specific cellular processes leave behind, such as metabolic intermediates or signaling molecules. Those small molecules are called metabolites. The metabolome of an organism is context-dependent and dynamic. Unlike for mRNA and proteins, it is difficult or impossible to establish a direct link between genes and metabolites without considering the physiological, developmental and pathological state of a cell. By detection, identification and quantification of single metabolites or patterns of metabolites one can examine the temporal changes caused by different factors such as nutrition, diseases, pharmaceuticals, or genetic effects.

Due to the different contexts metabolomics emerged from (profiling in plants or for clinical application) and based on the differences in metabolite coverage (targeted or untargeted analysis), accuracy, and instrumentation, several additional terms are in use for metabolomics: *metabonomics*, *metabolic fingerprinting*, or *metabolite profiling*. Nowadays, these terms are used interchangeably.

The biological questions addressed in this field of research are endless. Here, we only want to name a few: In diagnostics 95%, of the clinical essays test for small molecules [165]; in functional genomics, the direct functional information on metabolic phenotypes helps with annotation of gene function [35]; in drug discovery, many drugs are

¹<http://www.genomenewsnetwork.org/>

natural products or are inspired by them, such as antibiotics (e.g. penicillin), antiparasitics (e.g. avermectin), antimalarials (e.g. quinine), or anticancer drugs (e.g. taxol) [91]. In addition, new questions are raised continuously, for example, while studying microalgae that are of ecological and climate relevance. In the metabolomes of those organisms, unidentified metabolites are by far dominating the list of statistically relevant hits [159]. Identification and structural elucidation of such unknown metabolites is a major challenge in metabolomics.

Many researchers argue that metabolomics is still in its childhood [24]. Actually that is not quite true: for example, already in Traditional Chinese Medicine and Ayurvedic Medicine (1500–2000 BC) high doses of glucose were an indicator for diabetes. Here, “metabolite screening” was performed by ants or other insects attracted by a high dose of glucose, or by tasting of the urine [156]. Further, both Traditional Chinese Medicine and Ayurvedic Medicine used herbal medicine which is nothing else than plant (secondary) metabolites used as bioactive components for treatment of diseases. It is true, however, that the systematic profiling of metabolites in a high-throughput manner gained broad interest only during the last decade [24], partly because the analytical methods improved. Less systematic approaches using gas chromatography have been already proposed during the 1970s [56, 108]. The term “metabolic profiling” was introduced by Horning and Horning [56] in 1971 who used gas chromatography-mass spectrometry to measure compounds present in human urine and tissue extracts.

Different from the other “omic” approaches, it is currently not possible to analyze the entire range of metabolites by a single instrumental platform. Different technologies complement each other [24], Nuclear Magnetic Resonance (NMR) and Mass Spectrometry (MS) being the predominant ones. With increasing sensitivity, the utility of NMR to detect metabolites has improved making it a leading analytical tool to provide detailed structural information. But still, NMR is orders of magnitude less sensitive than MS [79]. In MS, the rapid advances in instrumentation made the method capable to perform high-throughput analysis. The amount of data produced is very hard to process and analyze manually [65]. MS is usually coupled to a separation method, in which gas chromatography-MS is one of the oldest techniques for metabolite profiling [56]. Here, the molecule is fragmented using electron ionization (EI) and masses of the fragment ions are recorded, revealing certain information about the molecular structure.

A major challenge in metabolomic analysis is the low proportion of detected analytes with unambiguously assigned chemical structures. As Patti *et al.* [107] mention in their 2012 review, “an astounding number of metabolites remain uncharacterized with respect to their structure and function”. The term “structure elucidation” usually refers to full *de novo* structure identification of an unknown organic compound, including stereochemical assignments. Structural elucidation is complicated by the high physical and chemical divergency of metabolites. Unlike biopolymers such as proteins and glycans, metabolites are not made up of repeated building blocks. The genome sequence does not reveal information about metabolite structure, as it does for protein structure. Structure confirmation is always performed with a set of independent methods. It is commonly believed that structure elucidation is impossible using MS techniques alone. Information from MS analysis can, however, strongly reduce the search space and give hint to the structure or class of the compound.

1.2 Contribution of this Work

In this work, we present the automated analysis of high resolution EI fragmentation mass spectra. MS experts evaluate fragmentation mass spectra by drawing fragmentation diagrams. For this task, the MS expert usually has to know the molecular structure of the compound. We present a method, that is extracting such information directly from the data, independent of existing library knowledge and without information about a compound’s structure. By automated signal extraction and evaluation, we explain relevant fragmentation reactions and assign molecular formulas to fragment ions. The method enables the identification of the molecular ion and the molecular formula of a metabolite if the molecular ion is present in the spectrum. This works even if the molecular ion is of very low abundance or hidden under contaminants with higher masses.

The annotation of fragmentation reactions and fragment formulas can be further used to compare compounds. Different from spectral library searching this allows for the detection of not only identical but also similar compounds. For large databases, these comparisons have to be performed very often. We present three algorithms for the problem and show that thousands of such comparisons can be computed in a matter of minutes.

The presented methods will be very helpful in the automated analysis of metabolites that are not included in common libraries and thus have the potential to support the explorative character of metabolomics studies. Several applications are possible: for example in drug research, screening of natural products is a difficult effort with a high probability of duplications [91]. Our methods may help with the dereplication of compounds at an early stage of the drug discovery process, that is, the detection of molecules that are identical or highly similar to known drugs or drug leads. Furthermore, when a potential drug lead has been determined, our approaches may help to identify it and elucidate its structure. Another example is the investigation of plankton interactions that are drivers for the global climate functioning. Filling the blanks in the metabolic maps of those microalgae may leads to the identification of fundamentally new communication and interaction mechanisms in nature [40].

To present the novel methods for the analysis of fragmentation mass spectra, this thesis is structured as follows: Chapter 2 introduces the main concepts of mass spectrometry-based metabolomics. Chapter 3 covers the computational aspects of identifying small molecules, from the identification of a compound searching a reference spectral library, to the concept of fragmentation trees, which allows a true *de novo* analysis of fragmentation data.

In Chapter 4 we describe a method for the automated fragmentation analysis of high resolution EI mass spectra based on a fragmentation tree algorithm. We focus primarily on the handling of hard ionization issues, mainly the identification of the low abundant molecular ion. Our method simultaneously identifies the molecular ion peak and molecular formula of an unknown compound and further computes a fragmentation tree that offers a hypothetical interpretation of the experimental data. We evaluate the identification of molecular ions and molecular formulas on two different datasets and discuss the capability of fragmentation trees to reconstruct fragmentation processes.

For the automated comparison of the fragmentation patterns of small molecules, Rasche *et al.* [113] introduced local fragmentation tree alignments. Aligning fragmentation trees is computationally hard. In Chapter 5, we present three exact algorithms for the problem: a dynamic programming algorithm, a sparse variant of the dynamic programming algorithm, and an Integer Linear Program. Evaluation of our methods on three different datasets

showed that thousands of alignments can be computed in a matter of minutes using dynamic programming, even for “challenging” instances. In addition, we demonstrate how to cluster compounds based on fragmentation tree similarities. This real-world example has been already presented by Rasche in his thesis [111] and is recapitulated here only to complement the chapter.

Finally, Chapter 6 concludes the thesis by recalling the main results and presenting an outlook on further applications of fragmentation trees and fragmentation tree alignments for the identification of unknown metabolites.

2 Biological Background, Analytical Concepts and Theoretical Notation

In this chapter we give a brief insight into the biological, analytical, and theoretical concepts that are required to understand this thesis. First, we present the biological unit we are interested in, namely small molecules, also called metabolites. Second, we introduce mass spectrometry – the analytical platform that is used to investigate these molecules. In addition to describing the basic components of mass spectrometry, we outline the main technical setups and the type of data generated by these methods which is analyzed in this work. Finally, we introduce the basic graph theoretical notation that is used throughout the thesis.

For all three research fields, we can cover only the aspects most important for this work. We refer interested readers to relevant textbooks: Weckwerth [162] for metabolomics, Gross [43] for mass spectrometry, and Diestel [23] for graph theory.

2.1 Molecules

Atoms are the basic building blocks of matter that cannot be decomposed chemically. Atoms consist of a dense central nucleus which contains protons and neutrons and is surrounded by electrons. The number of protons determines the chemical element of an atom. The number of neutrons determines the *isotope*; different isotopes of the same element have the same chemical properties but different mass. When a chemical symbol is used, e.g., C for carbon, standard notation is to indicate the number of *nucleons* (both protons and neutrons) with a superscript at the upper left of the chemical symbol, e.g., ^{12}C for the most abundant carbon isotope. If an atom contains an equal number of protons and electrons, it is electrically neutral, otherwise it is positively or negatively charged. Due to electromagnetic forces, atoms with opposing charges attract each other.

Electrons are organized in orbitals around the nucleus. Each orbital can contain only a fixed number of electrons. An orbital must be completely filled before electrons can be added to an outer one. Possessing completely filled orbitals is energetically optimal. Atoms of some elements can reach this state by forming chemical bonds. If both atoms equally share electrons to fill their highest orbital, the chemical bond is referred to as a *covalent bond*. If the electrons are bound more tightly to one of the atoms an *ionic bond* is formed.

A group of atoms connected by bonds is a *molecule*. Molecules are electrically neutral. An *ion* is an atom or molecule which has lower or higher number of electrons than protons and so possesses a positive or negative electrical charge, respectively. A charge can be given to a neutral molecule by adding or removing one or more electrons, by adding a positively charged ion (a cation), or by adding a negatively charged ion (an anion). Since electrons are paired up when chemical bonds are formed, most neutral molecules carry an even number of electrons. A *radical* is an atom, molecule, or ion that has unpaired

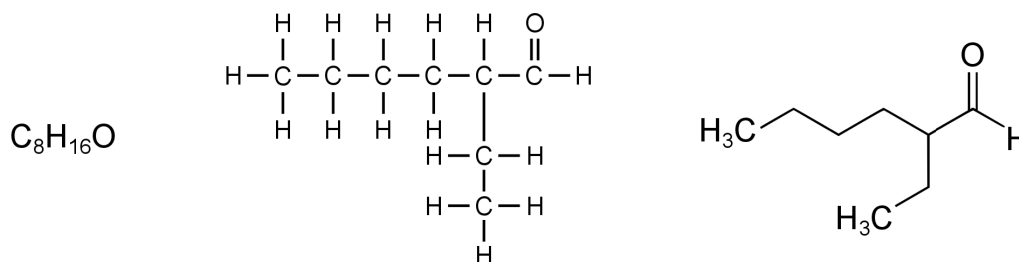


Figure 2.1: Molecular formula (left), Lewis structure (middle), and skeletal formula (right), of 2-ethylhexanal, a disinfectant and solvent from the group of aldehydes. Throughout the thesis, we use skeletal formulas to depict molecular structures.

electrons in its outermost orbital. Thus, a radical may be seen as an atom which has one or more pending covalent bonds and is therefore highly chemically reactive.

In solids and liquids molecules are often bound to other molecules by forces that are much weaker than covalent bonds, such as hydrogen bonds or van der Waals interactions. The resulting chemical structures are referred to as *chemical compounds*.

The *molecular mass* of molecules is expressed in *Dalton* (Da), or equivalently in *unified atomic mass units* (u). By definition, one Dalton is $1/12$ the mass of one atom of the ^{12}C isotope (which is $1.660538921 \times 10^{-27}$ kg). The total number of protons and neutrons of an ion or molecule is called the *nominal mass* or *nucleon number*. The calculated *exact mass* of an ion or molecule is obtained by summing the masses of all atoms using an appropriate degree of accuracy. Note that *molecular weight* is not the same as molecular mass, but is a ratio relative to ^{12}C which can vary with the geographical location [30]. It is not measured but typically just calculated from a particular chemical formula, using the average masses of the chemical elements in the formula.

The *molecular formula* indicates the exact number of atoms of each chemical element that compose a molecule. This elemental composition determines the mass of the molecule. Different molecular formulas can have the same nominal mass. The *structural formula* also represents the arrangement of the atoms relative to each other (see Figure 2.1). For example, the widespread Lewis structure is a flat graphical formula displaying the atoms as their elemental symbols and drawing lines between them to represent bonds. This notation is mostly used for small molecules. Different structural formulas can have the same elemental composition.

2.2 Metabolites

Metabolites are the intermediates and products of metabolism, that is, all life-sustaining chemical reactions within the cells of living organisms. The term metabolite is usually restricted to small molecules typically below 1000 Da. The structural diversity of metabolites is extraordinarily large in spite of their small size [35, 79]. They cover a wide array of compound classes, including sugars, acids, bases, lipids, hormonal steroids, and many others [21, 65]. Hence, the physical and chemical properties of metabolites are highly divergent [35]. Larger molecules that are made up of repeated building blocks, such as proteins and glycans, are not considered metabolites. Also, the structure of metabolites usually cannot be deduced by using genomic information. A notable exception are polyketides.

Metabolomics is a rapidly developing field of ‘omics’ research dealing with the detection, identification and quantification of metabolites. Traditionally, metabolites are divided into primary and secondary metabolites. Primary metabolites, directly involved in growth, development, and reproduction, have been thoroughly investigated as the basic metabolic pathways and components are similar even between vastly different species [104]. In contrast, secondary metabolites are often specific to a narrow set of species. All organisms synthesize huge numbers of secondary metabolites, but “an astounding number of metabolites remain uncharacterized with respect to their structure and function” [4]. Secondary metabolites, for example antibiotics or pigments, are not directly involved in the basic metabolic pathways, but usually have important ecological function.

The analysis and identification of small molecules are important in many areas of biology and medicine. For example, newly identified metabolites often serve as leads in drug design [91, 127], in particular for antibiotics.

2.3 Mass Spectrometry

Mass Spectrometry (MS) is a dominant technology for high-throughput analysis of metabolites and other small molecules [18, 35, 85]. It has excellent compound specificity and high sensitivity. In particular, MS sensitivity is orders of magnitude higher than that of nuclear magnetic resonance (NMR) [79].

Most mass spectrometers consist of three basic components (see Figure 2.2). An *ion source* to produce a gaseous state and to give the molecules charge; a *mass analyzer* to separate the ions according to their mass-to-charge ratio (m/z); and a *detector* to detect arriving ions. The ion detector produces an electric current proportional to the amount of ions it detects. Ions are neutralized when they collide with an earthed metal plate in the detector. In one type of detector, the *electron multiplier* [2], this plate is called a *dynode* and is coated with a material that emits secondary electrons when an ion collides with it. In a perfect detector, the signal intensity would be directly proportional to the amount of ions coming into contact with it. Although realistic instruments cannot provide this proportionality for all masses, they nevertheless have a linear range. Other ion detectors include *Faraday cups* [12], *photomultipliers*, and *micro channel plates*.

The resulting *mass spectrum* is a two-dimensional plot of signal intensity versus mass-to-charge ratio that typically consists of a series of *peaks* corresponding to detected ions. The mass-to-charge ratio m/z is dimensionless by definition, but the unit *Thompson* (Th) is sometimes used for this ratio in mass spectrometry. Two important characteristics of an MS measurement, and in general of MS instruments, are mass accuracy and mass resolution. *Mass accuracy* is the ratio of the m/z measurement error to the true m/z and is usually given in parts per million (ppm). *Resolution* measures the ability to distinguish two peaks of slightly different m/z . Resolution is typically given as *full width at half maximum* (FWHM). High resolution is immensely helpful for high mass accuracy [121] since it limits errors coming from inaccurate determination of signal centroids in unresolved peaks with close m/z .

The ionizers and mass analyzers relevant to this thesis are described below.

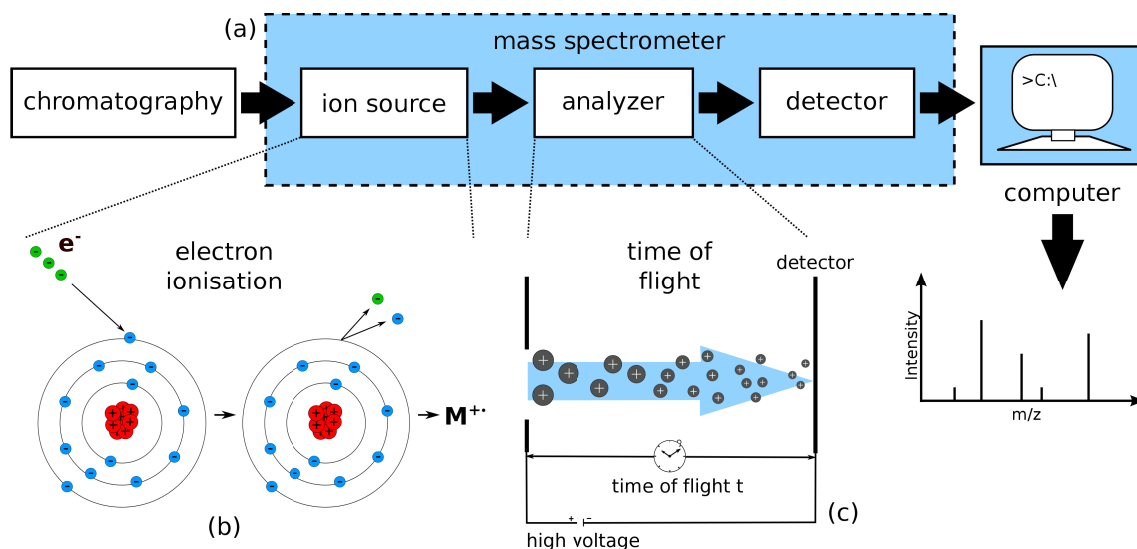


Figure 2.2: Schematic drawing of a mass spectrometer (MS). (a) Most mass spectrometers consist of three basic components: an ion source to give the molecules charge; a mass analyzer to separate the ions; and a detector to detect arriving ions. To analyze complex mixtures MS is often coupled to chromatography, such as gas chromatography (GC). (b) The most common ionization technique for GC-MS is electron ionization. Electrons are accelerated and knock electrons out of the outer orbitals of the gaseous sample molecules (for ease of presentation, a single atom is shown here). The radical cation formed is called molecular ion $M^{+\bullet}$. (c) GC is often coupled to time-of-flight analyzers. Ions are separated during their flight along a field-free path: smaller ions have higher velocities and fly faster than bigger ions.

2.3.1 Gas Phase Ion Generation Techniques

For the mass-to-charge ratios of molecules to be measured, they have to carry a net charge and be present in gaseous form. A charge can be applied to a neutral molecule by adding or removing one or more electrons, by adding a cation, or by adding an anion. Depending on the amount of energy that is transferred to the molecules during ionization, the ionization technique either induces little or no fragmentation (*soft ionization*), or a lot fragmentation (*hard ionization*).

Electron Ionization (EI), sometimes also called electron impact ionization, is the most widely used ionization method for metabolomics [36]. The sample (liquid or solid) must first be vaporized by heating to generate gaseous molecules. These are then exposed to a beam of electrons. Usually, gas chromatography (GC) has been performed beforehand (see Section 2.4.1). Clearly, only thermally stable compounds are amenable to EI, restricting this technique to a limited number of molecules.

In this method, electrons are accelerated by dropping them to a lower electrical potential. The fast moving electrons in the resulting electron beam carry about 70 eV energy and knock electrons out of the outer orbitals of the gaseous sample molecules (see Figure 2.2). The radical cation formed is called *molecular ion* $M^{+\bullet}$. Its molecular formula remains the same but the mass differs by the mass of one or more electrons. The ions generated by EI bear a lot of internal energy (“hot ions”) causing the bonds to break. Thus, EI is a hard

ionization technique. The molecular ions of labile molecules often have lifetimes shorter than 1 microsecond and are therefore not detected.

Electrospray Ionization (ESI) converts solution-phase compounds into gas-phase ions [33]. ESI can be easily coupled with liquid chromatography (LC) to separate complex mixtures online. ESI is a soft ionization technique, i.e., the generated ions carry very little internal energy ("cool ions"). Thus, ESI is a soft ionization technique. For this reason, an additional fragmentation technique is required (see Section 2.4.2).

The liquid sample is infused into a metal capillary that is held at an electric potential. At its tip, molecules become charged and the solution is emitted in a fine mist of droplets. As solvent evaporates from the charged droplets, surface charge is increased, which forces the droplets to explode. Small molecules are thought to be transferred into gas phase via the *ion evaporation model* (IEM) [81]: the ions evaporate from the solvent due to the field difference. In positive mode, the resulting ions typically carry an additional proton $[M + H]^+$. Larger molecules, such as biopolymers (DNA, proteins), often receive multiple charges.

2.3.2 Mass Analyzer

A mass analyzer separates ions according to their mass-to-charge ratio m/z . While the ionization method determines the class of molecules that are amenable for analysis, the mass analyzer (in combination with the detector) determines the quality of the measurement, that is, mass accuracy and resolution [171]. There are several types of mass analyzers, varying in physical principles and performance standards.

Time-of-Flight Mass Analyzer The principle of a Time-of-Flight (TOF) instrument is quite simple (see Figure 2.2): Ions of different mass-to-charge ratios, accelerated by the same electric force, acquire different velocities. Hence, ions with differing m/z can be separated during their flight along a field-free path. Smaller ions have higher velocities and fly faster than larger ions. This simple principle leads to straightforward design and construction making TOF instruments relatively inexpensive. In addition, they deliver a high acquisition rate of spectra [43].

In TOF we do not scan any parameters. The travel time t is proportional to the square root of m/z . This causes Δt for a given $\Delta m/z$ to decrease with increasing m/z [43]. Although the principle behind TOF instruments is quite old, it was not possible to put the idea into practice until computers were fast enough to measure the small time differences involved. This also explains the poor resolution in TOF instruments. High acceleration and long flight tubes aid in achieving more accurate spectra by increasing the differences between different mass-to-charge ratios. The *TOF reflectron principle* can improve resolution by focusing ions of different kinetic energies into one ion packet.

Orbitrap In the Orbitrap, accelerated ion packets are trapped in a stable trajectory around a spindle-shaped inner electrode. While spiraling around this electrode, ion packets of equal m/z have a characteristic frequency of axial oscillation, inducing an image current in two metal plates located near the electrode. A Fourier Transform is performed on the overlapping image current signals to obtain individual frequencies.

Orbitraps require little maintenance and have a small and compact design. They have a wide dynamic range, high mass accuracy and high resolution and sensitivity.

Quadrupole Filter A quadrupole filter consists of four hyperbolically or cylindrically shaped metallic rods. Each opposing pair is held at the same potential that is composed of a direct current (DC) component and an alternating current (AC) component, producing a highly oscillating electromagnetic field between them. When an ion enters the quadrupole it spirals between the rods with a radius depending on its m/z and the AC frequency of the field. For a fixed AC frequency, only ions within a particular m/z range will traverse through the quadrupole without hitting the rods. Quadrupole filters are fast and cheap, but can usually only operate at unit resolution.

2.4 Common Fragmentation Mass Spectrometry Setups

Several kinds of set-up have been developed for the mass spectrometric analysis of small molecules. To analyze complex mixtures, such as cell extracts, the MS instrument is often coupled to a separation device. The separation of the different components of a mixture is necessary to obtain compound-specific fragmentation spectra. The sample is dissolved in a mobile phase which is then forced through a column filled with a sorbent covered with a stationary phase. Separation is based on the differing interactions between the molecules and the two phases. These interactions cause the compounds to elute at different *retention times*. Retention times are system-dependent.

In column chromatography, such as gas chromatography, liquid chromatography or capillary electrophoresis, the stationary phase is placed in a narrow tube, through which the mobile phase is forced. This column can be easily coupled with a mass spectrometer.

To obtain information beyond the molecule mass, the analyte is usually fragmented, and masses of fragment ions are recorded. Depending on the way in which gaseous ions are generated, they carry different amounts of internal energy. For *hard ionization* methods such as EI, the transferred energy is already enough to fragment the ion. By contrast, the ions generated by *soft ionization* methods such as ESI, have very little internal energy. For these ions, an additional fragmentation technique is used. Typically, gas chromatography MS uses electron ionization (EI) fragmentation, whereas liquid chromatography ESI-MS is combined with *collision-induced dissociation* (CID). Fragmentation of a singly charged ion results in a fragment ion which retains the charge, and a *neutral* or *radical loss* that cannot be detected by the mass spectrometer.

In the following, the two “standard” experimental methods for small molecule analysis are described (see Table 2.1).

2.4.1 Gas Chromatography Electron Ionization Mass Spectrometry

Gas chromatography coupled to mass spectrometry (GC-MS) is extensively used in metabolome analysis, and was widespread decades before liquid chromatography-MS (LC-MS) [36, 56]. GC is arguably still the best separation tool in common use for compounds amenable to the technique [26]. In GC, the mobile phase is usually an inert carrier gas, such as helium or nitrogen, which carries the sample through the column. Gaseous compounds interact with the walls of the column, which is coated with a stationary phase, usually a microscopic layer of high-boiling-point liquid. Different interactions cause the

Table 2.1: Comparing the two “standard” experimental setups for fragmentation MS of small molecules. Gas chromatography electron ionization mass spectrometry (*EI mass spectra*) was widespread in metabolomics, decades before liquid chromatography collision-induced dissociation mass spectrometry (*tandem mass spectra*). The two methods complement each other in terms of amenable molecules and quality of the resulting mass spectra.

EI mass spectra		tandem mass spectra	
cons	only amenable to volatile/thermally stable compounds	pros	broad range of compounds
	mass of the unfragmented molecule unknown		mass of the unfragmented molecule known
	primarily low mass accuracy		often high mass accuracy
pros	fragment-rich spectra	cons	several CEs have to be applied to get fragment-rich spectra
	fragmentation processes well understood		fragmentation not completely understood
	highly reproducible		less reproducible across different instruments/instrument types

compounds to elute at different retention times. GC enables the separation of more than 100 compounds in a single run [118].

The most common ionization technique for the analysis of small molecules by GC-MS is *electron ionization* (EI), which enables easy interfacing of GC with MS. EI is the oldest ionization technique and its fragmentation mechanisms are well described [96]. Having already been converted to gaseous form for GC separation, the molecules ionize and fragment as they are exposed to a beam of free electrons (see Section 2.3.1). Because of the constant ionization energy at 70 eV, which is much higher than any covalent bond, the resulting mass spectra are in general consistent across instruments and specific for each molecule [78, 79]. This permits comparisons with library entries or spectra of known standards [82, 98]. The high ionization energy transfers a lot of energy to the ions. The resulting spectra are fragment-rich but often also show a low or missing molecular ion peak; to this end, the mass of the compound is often unknown.

Historically, GC has primarily been coupled to MS instruments that provide only nominal mass information, such as quadrupole instruments or, at best, relatively bad mass accuracy (worse than 100 ppm, parts per million). However, this is not a fundamental problem of GC-MS. More recently, GC has been coupled to high-resolution TOF instruments, providing high mass accuracy measurements [15, 49, 61].

GC-MS requires a volatile and thermally stable analyte. Naturally occurring volatile metabolites have a boiling point lower than 300°C. Chemical derivatization enables the analysis of many semi-volatile compounds by decreasing their boiling points and protecting them against thermal degradation. In addition, selective derivatization can help with the detection of functional groups [44]. For organic compounds, such as alcohols or phenols, silylation is the most widely used derivatization procedure. An active hydrogen is replaced by an alkylsilyl group, usually trimethylsilyl (TMS)

[$-\text{Si}(\text{CH}_3)_3$] (see Figure 4.3 for an example). The resulting derivatives are generally more volatile, less polar, and more thermally stable than their precursors. However, silylation is not simple, fast or easy to automate. Alkylation is the most used technique for derivatization of polyfunctional amines and organic acids [160]. Here, an active hydrogen is replaced by an aliphatic or aliphatic-aromatic group. Again, the resulting derivatives are less polar and more volatile. For metabolomics, silylation using N-methyl-trimethylsilyltrifluoroacetamid (MSTFA) [44] and alkylation using O-(2,3,4,5,6-pentafluorobenzyl)-hydroxylamine hydrochloride (PFBHA) [80] are routinely used in GC-MS analyses.

In the following we refer to mass spectra from this experimental setup as *EI mass spectra*.

2.4.2 Liquid Chromatography Collision-induced Dissociation Mass Spectrometry

In order to overcome the drawback of only being able to analyze thermally stable compounds, *liquid chromatography* MS (LC-MS) has become increasingly used for the analysis of small molecules. Initially, there were major difficulties in coupling LC with MS [160]. A solution to these difficulties was provided by the introduction of the more gentle *electrospray ionization* (ESI) technique [100].

As no thermal volatilization is necessary for LC, it is amenable to a much wider range of molecules, including many secondary metabolites [35]. LC requires only small amounts of material [107] and as no derivatization step is required, sample preparation is simple [160]. The liquid mobile phase is, typically, a mixture of solvents, and the stationary phase is a porous solid. The complex physical interactions between sample components and the solid phase influence the retention times of the individual compounds. Originally the mobile phase flowed through the stationary phase under the force of gravity alone. The use of *high pressure* pumps (HPLC) has increased flow rates and separation efficiency as much smaller particles can be used.

The ionization technique of choice for LC-MS based metabolomics is ESI [21]. In contrast to other ionization methods (e.g. MALDI), here the low m/z range is less obscured by chemical noise [81]. ESI is a soft ionization method, which results in minimal fragmentation of the ions. This has the advantage that the mass of the unfragmented analyte can be recorded. In a second step, a selected compound is fragmented in a collision cell, resulting in a fragmentation spectrum. The ion selected for fragmentation is called the *precursor ion*. The whole experimental setup is called *tandem MS* (MS^2).

In metabolomics, compounds are most commonly fragmented using *Collision Induced Dissociation* (CID). The collision cell is filled with an inert gas (nitrogen or a noble gas, such as argon). Sample ions are accelerated and collide with the inert gas molecules resulting in fragmentation. The acceleration of the ions, and thus the intensity of the collisions, can be adjusted. This *collision energy* is measured in electron volts (eV) and typically ranges from 5 to 100 eV. Tandem mass spectra usually contain far fewer fragments than EI fragmentation spectra. To increase the number of fragments, several spectra are recorded, each captured at a different collision energy. Alternatively, *CID voltage ramping* continuously increases the fragmentation energy during a single acquisition [41]. Due to the complex rules of the gas phase chemistry responsible for fragmentation, the understanding of CID fragmentation is still in its infancy for metabolites. Even at high energies, CID often leads to only poor fragmentation [103]. *Higher-energy Collisional Dissociation* (HCD) usually results in a larger diversity of fragment ions.

Unfortunately, CID mass spectra are less reproducible than EI spectra, particularly across different instrument types or even instruments [16]. When comparing spectra from different instrument types, only 64–89 % (depending on the instruments) of the spectra pairs match with more than 60 % identity [11]. This complicates the otherwise simple task of searching spectral libraries [101]. Using different collision energies makes spectra even harder to compare. Some progress has, however, been made in normalizing fragmentation energies across instruments and instrument types [16, 53, 106].

In the following we refer to mass spectra from this experimental setup as *tandem mass spectra*.

Multiple-stage MS (MS^n) allows expansion of the information obtained by LC-MS with additional fragmentation reactions. To this end, ions corresponding to several peaks from the initial fragmentation step can be selected (manually or automatically) and subjected to another fragmentation reaction. The resulting fragment ions can, in turn, again be selected as precursor ions for further fragmentation. This increases the number of fragments even more and gives information about dependencies between them. Typically, with each additional fragmentation reaction, the quality of mass spectra is reduced and measuring time increases. Thus, analysis is normally limited to only a few fragmentation reactions beyond MS^2 .

Tandem MS and multiple MS are often performed on instrumental platforms that result in high mass accuracy spectra, such as Orbitrap.

2.4.3 Further Experimental Setups

Besides the two common fragmentation MS setups described above, other setups, for example using alternative ionization techniques [95, 150], have been developed. We can classify these different setups by referring to the characteristics of the two standard setups: For example, is the mass of the molecular ion known (LC- MS^2) or unknown (GC-EI-MS)? Is the fragmentation spectrum rich (GC-EI-MS) or sparse (LC- MS^2)? A good computational MS method does not target only one particular experimental setup but can be adapted, with little effort, to other systems.

2.5 Graph-Theoretical Notation

In this thesis we analyze small molecule fragmentation MS data using graphs. Here, we introduce the basic terminology and notation from graph theory. For further information on graph theory, see Diestel [23].

Definition 1 (Graph). An *undirected graph* $G = (V, E)$ is a pair of a vertex set V and an edge set $E \subseteq \binom{V}{2}$, where $\binom{V}{2}$ denotes the collection of all two-element subsets of V . An *undirected edge* $e = \{u, v\}$ connects vertices u and v . If the order of the vertices of every edge in G is fixed, we say that G is a *directed graph*. In a directed graph, edges are ordered pairs $e = (u, v)$ and hence $E \subseteq V \times V$. We often use $e = uv$, instead of $e = (u, v)$ in directed graphs for brevity.

In this work we deal with directed graphs. We label vertices and edges with molecular formulas that are multisets of elements. This differs from the usual meaning of “labeled graph”, in which only the vertices are labeled, and these labels are required to be unique.

Definition 2 (Weighted Graph). If there is a function $w : E \mapsto \mathbb{R}$ defined on the edge set, a graph $G = (V, E)$ is *edge-weighted*. We call $w(e)$ the *weight* of the edge e . Analogously, if a function $w_V : V \mapsto \mathbb{R}$ exists, we call G *vertex-weighted*.

We weight graphs to distinguish between meaningful and less important edges and vertices.

Definition 3 (Colored Graph). Given a set of colors C , if there is a function $c : V \mapsto C$ a graph $G = (V, E)$ is *vertex-colored*. We call $c(v)$ the *color* of vertex v .

Often, color is used to denote the type of a vertex, or, in our case, vertices that share the same origin.

Definition 4 (Colorful). A graph is *colorful* if every color occurs at most once in the graph, that is, every vertex possesses a unique color.

Definition 5 (Subgraph). A graph $G' = (V', E')$ is a *subgraph* of the graph $G = (V, E)$ iff $V' \subseteq V$ and $E' \subseteq E$.

A *tree* is an undirected simple graph G that is connected and has no cycles. A directed graph which contains no directed cycles is called a *directed acyclic graph* (DAG). It represents a hierarchy of objects. We call a DAG *transitive* if its edge relation $E \subseteq V \times V$ is transitive. An *arborescence* is a DAG that does not contain cycles even if its edges were considered to be undirected and whose edges all point away from a particular vertex called the *root*. For simplicity, we call arborescences *trees* throughout this thesis. In such trees, the *parent* of a vertex v is the vertex u that is connected to v by a directed edge uv . Every vertex except the root has only one incoming edge and therefore a unique parent. A *child* of a vertex v is a vertex of which v is the parent. The *outdegree* of a vertex is the number of outgoing edges, that is, the number of children.

Vertices are also called *nodes*. We will use this term throughout the thesis.

3 Computational Analysis of Small Molecule MS Data

In recent years, it has been recognized that the major bottleneck in small molecule MS is the automated processing of the resulting data [98]. The amount of data produced during metabolomic analysis is hard to process and analyze manually [65]. Such manual data analysis requires not only a lot of time, but also deep knowledge of the underlying chemistry [163].

In this chapter we describe several computational methods for the analysis of small molecule fragmentation MS data, that is, EI mass spectra and tandem mass spectra. As it is outside the scope of this thesis, we will not describe computational methods that deal with the chromatography part of the analysis, such as predicting retention indices [37, 147]. Furthermore, we do not cover the problem of aligning two or more LC-MS or GC-MS runs [14, 67, 92, 138].

In the following, we will assume that the result of an MS measurement is a list of peaks, that is, pairs of m/z and intensity. In reality, this list is the result of several steps of processing the raw data, such as de-noising and peak picking; see Katajamaa and Oresic [72] for details.

For a comprehensive overview of experimental and computational techniques for small molecule mass spectrometry, from processing the raw data to structure elucidation, see Kind and Fiehn [78]. The basic computational approaches for dealing with small molecule fragmentation MS data are covered in [62, 63, 126].

3.1 Reference Data

A major problem for the development of novel algorithms is a lack of reference data. Unfortunately, the practice of making experimental data available is much less pronounced in the metabolomics and small molecule research community than it is in proteomics or genomics.

There exist two important commercial libraries: The National Institute of Standards and Technology (NIST) mass spectral library (version 11), which contains EI spectra of more than 200 000 compounds and collision cell spectra for about 4 000 compounds; and the Wiley Registry (9th edition), which comprises EI spectra of almost 600 000 unique compounds, as well positive- and negative-mode spectra of more than 1 200 compounds contained in the Wiley Registry of Tandem Mass Spectral Data [101, 102].

However, to allow data-driven development of algorithms for small molecule identification, mass spectrometric reference datasets must be made *publicly available* via reference databases, such as MassBank [54, 55], METLIN [137, 151], or Golm Metabolome Database (GMD) [83]. For tandem mass spectra, the attempts to make data publicly available were rather successful: METLIN [137] contains high resolution tandem mass spectra for more than 10 000 metabolites for diagnostics and pharmaceutical biomarker discovery [123] and

MassBank [54, 55] comprises more than 30 000 spectra of about 4 000 compounds collected from different consortium members. Unfortunately, for EI spectra, the size of publicly available reference databases remains small. For example, the GMD [83] contains EI fragmentation mass spectra of about 1 600 compounds. Furthermore, these spectra provide only nominal mass information, as GC has historically been coupled to MS instruments with relatively low mass accuracy (see Section 2.4.1). Even the commercial NIST (version 11) provides to a great extent only nominal mass EI spectra. Quite recently, GC has been coupled to instruments, providing high mass accuracy measurements [15, 49, 61]. However, obtaining high mass accuracy EI spectra of authentic standards remains difficult.

The restricted data sharing also prevents a comparative evaluation of methods. Recently, a first benchmark test for small molecule fragmentation data (both tandem mass spectra and *nominal mass* EI mass spectra) was provided as part of the CASMI challenge¹. Results are published in [128]; results of the fragmentation tree analysis (see Section 3.6) are published in [28].

3.2 Compound Identification and Structure Elucidation

The Chemical Analysis Working Group (CAWG) as part of the Metabolomics Standards Initiative (MSI) [94] established confidence levels for the identification of non-novel chemical compounds [148], ranging from level 1 for a rigorous identification based on independent measurements of authentic standards, to unidentified signals at level 4.

For novel and chemically uncharacterized metabolites, we have to overcome the boundaries of (spectral and molecular structure) databases. “Structure elucidation” usually refers to full *de novo* structure identification, including stereochemical assignments. It is commonly believed that structure elucidation is impossible using MS techniques alone, at least without using strong background information. Instead, structure confirmation of an unknown organic compound is always performed with a set of independent methods, in particular NMR. However, computational *de novo* methods for mass spectral data can strongly reduce the search space or give hints to the structure or class of the compound (see Sections 3.5 and 3.6).

3.3 Searching in Spectral Libraries

The usual approach for identification of a non-novel metabolite is to look it up in a spectral library. Library search requires a similarity or distance function for spectrum matching. A huge number of scorings (or similarity measures) have been developed over the years [143, 146]. Often, this is done using the “dot product” of the spectra, which can be improved by using a weight function to differently weight the terms of the product depending on the mass. The spectral dot product is an advanced form of the most fundamental scoring, namely the “peak counting” family of measures that basically count the number of matching peaks.

Searching in libraries of reference spectra provides the most reliable source of identification in case the library contains a fragmentation spectrum from a reference compound measured on a similar instrument [98]. Unfortunately, spectral libraries are vastly incomplete and only little progress has been made in establishing the confidence of

¹Critical Assessment of Small Molecule Identification, <http://casmi-contest.org/>

an identification [143]. False negative identifications occur if the spectrum of the query compound differs from the spectrum in the library, for example due to contaminations, noise (especially in low signal spectra), or different collision energies (CID). A reliable identification of a compound depends on the uniqueness of its spectrum. But the presence and intensity of peaks across spectra is highly correlated, as these depend on the non-random distribution of molecular (sub-)structures. This becomes a crucial problem when the database contains thousands of spectra. Unlike in proteomics, False Discovery Rates (FDRs) cannot be estimated as no appropriate decoy databases can be constructed. Usually, confidence in search results must be manually assessed by the user, based on the search algorithm used and the quality of spectrum and library [142]. Using fragmentation trees (see Section 3.6) as a detour in library searching allows us to compute such FDRs for small molecule MS.

EI mass spectra are, in general, highly reproducible even across instruments, and specific for each molecule [78, 79]. The Automated Mass Spectral Deconvolution and Identification System (AMDIS) [145] is the most commonly used free software for performing identification, and can identify huge numbers of metabolites that are cataloged in libraries. These libraries are often huge, since reference spectra have been collected over many years [143], but also commercial (see Section 3.1). However, where the compound is unknown, comparing the spectrum obtained to a spectral library will result in imprecise or incorrect hits, or no hits at all [36, 65, 78].

Fragmentation by tandem MS is less reproducible than EI fragmentation, in particular across different instrument types or even instruments [16, 101]. Only first steps have been taken towards searching tandem mass spectral libraries [102], and these libraries are much smaller than for EI mass spectra. Reliable library identifications can be achieved when a spectrum is acquired under the same conditions as the reference spectrum [69]. Attempts have been made to create more reproducible and informative tandem mass spectra [16, 41, 53].

When the true spectrum is not contained in the database, false positive hits may at least hint at correct “class identifications”. The NIST *MS Interpreter* [144] for EI mass spectra uses a nearest-neighbor approach to generate substructure information. A library search provides a list of similar spectra. Structural features of the unknown compound, such as aromatic rings or carbonyl groups, are deduced from common structural features of the hits. Demuth *et al.* [20] proposed a similar approach, and evaluated whether spectral similarity is correlated with structural similarity of a compound. Based on this evaluation, they proposed a threshold for spectral similarity that supposedly yields hit lists with significantly similar structures.

For a review on basic principles, practices, and pitfalls in the process of metabolite identification using spectral libraries, see Stein [143].

3.4 Molecular Formula Identification

One of the most basic — but nevertheless highly important — steps when analyzing an unknown compound, is to determine its molecular formula, often referred to as the “elemental composition” of the compound. Common approaches first compute all candidate molecular formulas (over a fixed alphabet of elements) that are sufficiently close to the measured peak mass [6, 8]. The six elements most abundant in metabolites are carbon (C), hydrogen (H), nitrogen (N), oxygen (O), phosphorus (P), and sulfur (S) [65]. In higher

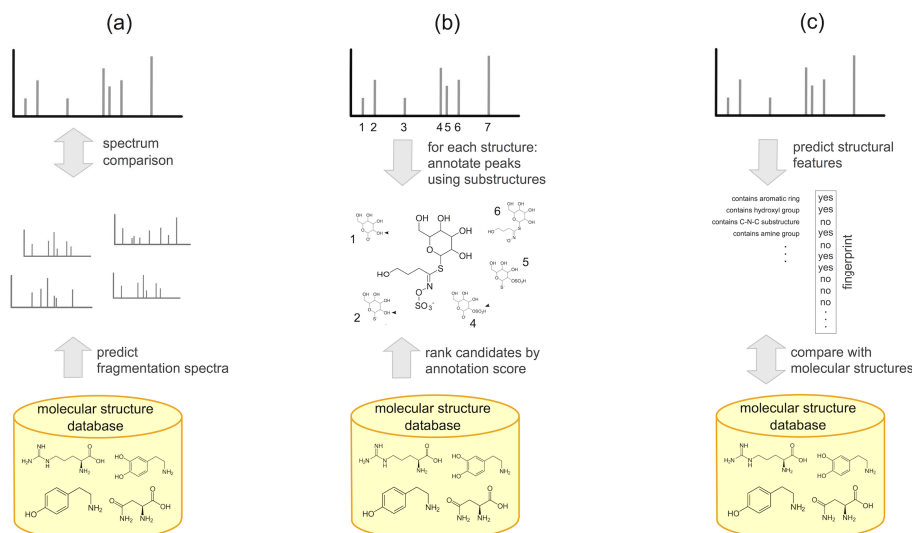


Figure 3.1: The three basic approaches of searching in molecular structure databases: (a) fragmentation spectrum prediction; (b) combinatorial fragmentation; and (c) predicting structural features.

mass regions there are too many candidate molecular formulas even with very high mass accuracy [76]. Kind and Fiehn [77] proposed “Seven Golden Rules” to filter molecular formulas based on chemical considerations. However, for larger masses, many molecular formulas pass these rules. For each candidate molecular formula, an isotope pattern is simulated [9, 17, 34, 117] and compared to the measured one [9, 109], to determine the best matching molecular formula. For this purpose, high mass accuracy is required and is nowadays available from a multitude of MS platforms. The molecular formula of the compound can serve as a basis for subsequent structure elucidation.

The molecular formula of an unknown compound can also be determined by computing fragmentation trees for all candidate molecular formulas [7]. In fact, fragmentation trees were initially introduced for this task (see Sections 3.6 and 4.2.2).

An overview of isotope pattern simulation is given by Valkenborg *et al.* [155].

3.5 Searching in Molecular Structure Databases

Spectral libraries are (and will always be) several orders of magnitude smaller than molecular structure databases, such as KEGG (Kyoto Encyclopedia of Genes and Genomes) and PubChem. For example, PubChem currently contains over 30 million pure and characterized chemical compounds. If desired, one can construct a fully comprehensive molecular structure database which comprises all feasible structures enumerated by molecular isomer generators [74]. Three recent approaches seek to replace searching in spectral libraries by searching in the more comprehensive molecular structure databases: (a) rule-based *in silico* fragmentation spectrum prediction; (b) mapping the fragmentation spectrum to a compound structure (combinatorial fragmentation); and (c) predicting structural features and compound classes (see Figure 3.1).

3.5.1 Rule-based Fragmentation Spectrum Prediction

The gap between molecular structure databases and spectral libraries may be filled by a database of theoretical fragmentation mass spectra predicted from molecular structure databases. A set of candidate molecules is generated by filtering a molecular structure database using the molecular mass of the unknown, or even its molecular formula if already known. Given this set of candidate molecular structures, spectra can be predicted by applying fragmentation rules to these structures.

Fragmentation rules are manually curated from mass spectrometry literature or can be automatically learned. First attempts at generating structural candidates and predicting their fragmentation mass spectra using general models of fragmentation, as well as class-specific fragmentation rules, were made as part of the DENDRAL project starting in 1965 [87, 88, 97, 139]. The DENDRAL project stopped after it became clear that automated structure elucidation using MS data could not be achieved at that time. Citing Gasteiger *et al.* [38]: “However, it is sad to say that, in the end, the *DENDRAL* project failed in its major objective of automatic structure elucidation by mass spectral data, and research was discontinued.” Nowadays, there are three major commercial tools that predict MS fragmentation based on rules: *Mass Frontier* (HighChem, Ltd. Bratislava, Slovakia; versions after 5.0 available from Thermo Scientific, Waltham, USA), *ACD/MS Fragmenter* (Advanced Chemistry Labs, Toronto, Canada), and *MOLGEN-MS* [73, 75].

Rule-based prediction systems were initially developed for the prediction and interpretation of EI mass spectra, which are highly reproducible. Much is known about fragmentation during EI, but complex rearrangements are relatively hard to predict. Schymanski *et al.* [129] compared the three commercial programs, and indicated that at the time of evaluation, mass spectral fragment prediction for structure elucidation was still far from daily practical use. The authors noted that *ACD Fragmenter* “should be used with caution to assess proposed structures [...] as the ranking results are very close to that of a random number generator.”

For tandem MS, the fragmentation behavior of small molecules under varying fragmentation energies is much less understood [163]. Nevertheless, there has been a recent tendency to investigate general fragmentation rules of tandem MS and interpret the data with rule-based prediction programs, too.

High-quality fragmentation prediction requires expert-curated “learning” of fragmentation rules. Even the best commercial systems cover only a tiny part of the rules that could be known. Although novel rules are constantly added, it is not necessarily the case that these rules will apply to a newly discovered compound. Moreover, for many rule-based fragmenters, all predicted peaks have the same intensity as bond cleavage rates are not considered. Accurate peak intensities can, however, greatly improve identification accuracy. Instead of curating or learning real fragmentation rules, Kangas *et al.* [70] use machine learning to find bond cleavage rates for spectral simulation. Doing so, cleavage rates and hence peak intensities can be estimated. Different from the rules learned for example during the DENDRAL project, they do not claim these predictions to be true fragmentation rules. Their *In Silico* Identification Software (*ISIS*) currently works only for lipids and does not model rearrangements of atoms and bonds.

It is worth mentioning that *in silico* fragmentation spectrum prediction has been very successfully used in proteomics for many years, as prediction of peptide fragmentation is comparatively easy. There, rule-based systems did not have much impact as it was

apparent from the beginning that, in view of the huge search space, only methods based on combinatorial optimization can be successful.

3.5.2 Combinatorial Fragmentation

In contrast to rule-based fragmentation prediction, combinatorial fragmentation attempts to explain the peaks in a measured spectrum by means of bond disconnections. This is based on the assumption that most peaks result from substructures of the compound without major rearrangement. Fragments resulting from structural rearrangements have to be individually “woven” into the combinatorial optimization. In fact, structural rearrangements are a problem for both combinatorial and rule-based methods [46, 51, 149].

The exhaustive enumeration of all fragments by applying all combinations of bond cleavages is very slow and hence cannot be applied for a large set of candidate molecular structures. Therefore, early approaches [46, 51] did not aim at finding a molecular structure but instead, explaining each peak in a fragmentation spectrum with the *most likely* substructure of a particular known molecular structure. To apply this approach to a set of candidate molecules filtered from a molecular structure database, faster methods to solve the underlying problem are required.

The most recent approach is *MetFrag* [166], a somewhat greedy heuristic that makes no attempt to create a mechanistically correct prediction of the fragmentation processes. A tree search algorithm is applied to enumerate possible fragments of the molecule: The root of the tree is the intact molecular structure, edges represent bond cleavages, and nodes are resulting fragments. To avoid combinatorial explosion, the number of cleavages is limited by a maximum tree depth. Further, redundant fragments and fragments smaller than the lightest fragment in the spectrum are removed. It is therefore fast enough to screen dozens to thousands of candidates retrieved from molecular structure databases, and to rank them by the agreement between measured and *in silico* fragments. We stress that *MetFrag* is not designed to explain a maximum number of fragments but rather to explain enough fragments to identify the compound in a molecular structure database. In fact, Wolf *et al.* [166] found that the prediction accuracy decreases with increasing maximum tree depth: the increasing number of simulated fragments also generates more unlikely fragments.

One problem of combinatorial fragmentation is choosing the costs for cleaving edges (bonds) in the molecular structure graph. In *MetFrag* the cost of cutting a fragment out of a molecule is the sum of bond dissociation energies of the cleaved bonds. However, Ridder *et al.* [116] report that even a simplistic scoring which basically assigns score 1 to single bonds, 2 to double bonds etc. outperforms the more involved cost function of *MetFrag*. This underlines that finding a suitable cost function remains an important open problem. Kangas *et al.* [70] propose machine learning to find bond cleavage rates for spectral simulation (see Section 3.5.1).

Recently, Gerlich and Neumann [39] introduced *MetFusion*, which combines *MetFrag* with spectral library search in MassBank to improve compound identification. In this way, *MetFusion* takes advantage of both available resources: molecular structure databases and spectral libraries.

3.5.3 Predicting Structural Features and Compound Classes

Learning from mass spectral data can be applied in many ways. Rather than predicting tandem mass spectra from molecular structures, machine learning can be used to predict

structural properties or compound classes from the spectra. The term *compound class* is not exactly defined: molecules may fall into the same group because they share a common reactive group, a substructure, have a certain chemical property, or a similar biological function. Usually, a mixture of these class types is used in application.

Given the spectrum of an unknown compound, a classifier gives a response telling us whether a particular substructure (or a more general chemical property) is present or not in the investigated compound. In its simplest form this is a *yes/no* answer, but alternatively some score or likelihood may be reported. To learn and predict structural properties from mass spectral data, spectra need to be transformed to a set of numerical *features* characterizing them. It has been observed very early, that appropriate transformation of the original spectral data is essential for a good prediction of structural properties [157]. The classifiers are trained on a set of feature vectors from mass spectra of known reference compounds. The feature vector of a query spectrum of an unknown compound is then given to the substructure classifiers to predict the fingerprint of the molecule; that is, a vector of *yes/no* answers indicating which substructures are present or not. Usually, the predicted fingerprints are directly used to characterize the class and properties of the measured metabolite. Going one step further, one can use the predicted fingerprints to retrieve and score candidate molecules from molecular structure databases [47].

The above idea was pioneered in 1969 [158] “to identify the general nature of the compound and its functional groups.” Kwok *et al.* [84] and Scott and coworkers [131–133] use pattern recognition to classify the unknown compound and class-specific rules are used to predict its nominal molecular mass or, more precisely, the mass difference to the detectable fragment peak of highest mass. The feature-based classification approach by Varmuza and Werther [157] for EI spectra uses a set of mass spectral classifiers to recognize the presence/absence of 70 substructures or general structural properties in the compound. This approach has found wide acceptance in the community as it is part of the NIST software. Much later, Hummel *et al.* [64] learned decision trees using mass spectral features and retention index information from the Golm Metabolome Database (GMD). Using these trees they predict frequent substructures and subdivide compounds into different compound classes.

Whereas the above methods are targeted towards EI mass spectra, the approach of Heinonen *et al.* [47] targets tandem mass spectra. Using a kernel-based approach the characterizing fingerprint of the unknown metabolite is predicted from the mass spectrum and matched against a molecular structure database.

3.6 Fragmentation Trees

Besides the prediction of structural features, not much progress has been made towards the *de novo* interpretation of fragmentation mass spectra of small molecules that cannot be found in any (not even a structural) database. This is the case for many metabolites which remain uncharacterized with respect to their structure and function [107]. In their 2010 review, Kind and Fiehn [78] state that “any *de novo* interpretation of such data is still challenging, if not totally impossible, due to the high molecular diversity and many similar compound structures”.

Given the molecular structure of a compound and the measured fragmentation spectrum, an MS expert can assign peaks to fragments of the compound and derive a “fragmentation diagram”. However, this is infeasible if we do not know the molecular structure, or when

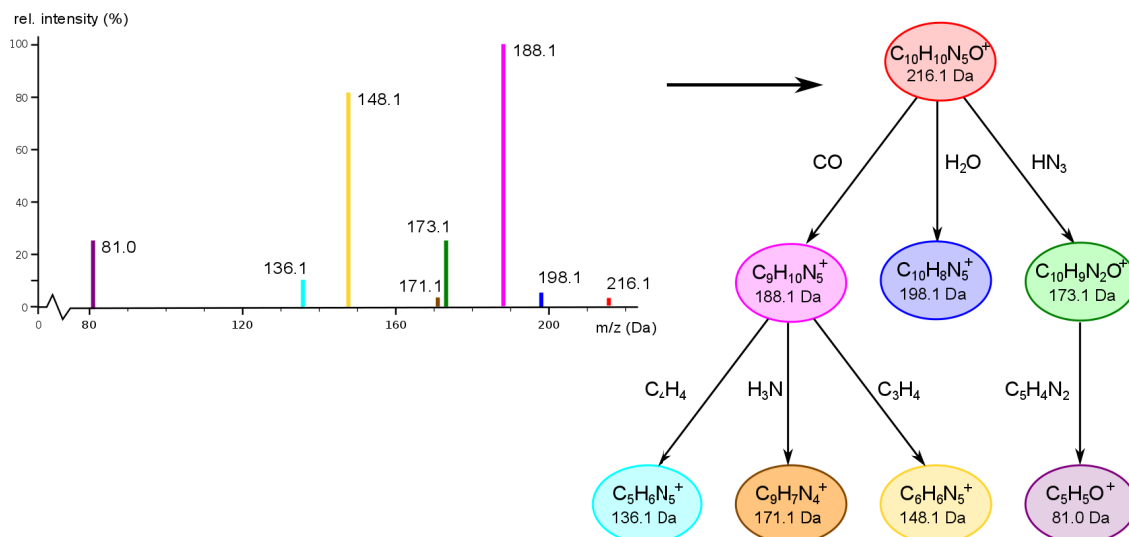


Figure 3.2: Fragmentation tree calculated by Rasche *et al.* [112] from a (merged) tandem mass spectrum of kinetin ($\text{C}_{10}\text{H}_9\text{N}_5\text{O}$), a plant hormone that promotes cell division. Each node is annotated with a molecular formula explaining the corresponding peak; edges are implicitly annotated with molecular formulas of losses.

thousands of spectra have to be analyzed. *Fragmentation trees* are similar to experts’ “fragmentation diagrams” but are extracted directly from the data, without knowledge about a compound’s structure. To compute a fragmentation tree, we need neither spectral libraries nor molecular structure databases; this implies that this approach can target “true unknowns”.

A fragmentation tree consists of nodes corresponding to the precursor ion and fragment ions, and directed edges connecting the nodes (see Figure 3.2). Each node is annotated with the molecular formula of the ion; edges are implicitly annotated with molecular formulas of losses. Given a fragmentation spectrum (possibly merged from several spectra at different energies) of an unknown compound, a fragmentation graph is constructed that contains all possible explanations for this spectrum, and a fragmentation tree is computed from the graph using combinatorial optimization. In the resulting tree, each node “explains” a peak in the measured fragmentation spectrum, that is, the mass difference between the node’s molecular formula and the observed peak mass is below the assumed mass accuracy. To this end, fragmentation trees introduce an “annotation layer” on top of the raw fragmentation data.

Böcker and Rasche [7] introduced fragmentation trees for tandem MS data to find the molecular formula of an unknown without using databases: here, the highest-scoring fragmentation tree for each molecular formula candidate is used as the score of the molecular formula itself (see Section 4.2.2). In 2011, Rasche *et al.* [112] found that for tandem mass spectra, fragmentation trees are reasonable descriptions of the fragmentation process and hence can also be used to derive further information about the unknown compound. Fragmentation trees can also be computed from multiple MS data [125].

For a given fragmentation spectrum, combinatorial optimization is used to find the tree that, according to some scoring function, *best* (and hopefully correctly) explains the observed spectrum. Unfortunately, this is impeded by the size of the search space: the fragmentation spectrum of a compound may be explained by numerous fragmentation trees.

Finding an optimal fragmentation tree has been proven to be computationally hard [114]. This severely complicates the design of swift algorithms for the problem. Algorithmic aspects of computing fragmentation trees are considered in [114].

To compare two unknown compounds based on their fragmentation spectra, Rasche *et al.* [113] introduced fragmentation tree alignments. By this, similar fragmentation cascades in the two trees are identified and scored. Fragmentation tree alignments can be used to cluster unknown compounds (see Section 5.4), to predict chemical similarity, and to find structurally similar compounds in a spectral library using *FT-BLAST* (Fragmentation Tree Basic Local Alignment Search Tool). *FT-BLAST* also offers the possibility to identify bogus hits using a decoy database, allowing the user to report results for a pre-defined False Discovery Rate.

Fragmentation trees must not be confused with *spectral trees* for multiple stage mass spectrometry [135], or the closely related *multistage mass spectral trees* of Rojas-Chertó *et al.* [119] (referred to as “fragmentation trees” in [71, 119, 120]). Spectral trees are a formal representation of the MS setup and describe the relationship between the MSⁿ spectra, but do not contain any additional information.

4 Fragmentation Trees for Electron Ionization Mass Spectra

When analyzing fragmentation spectra of small molecules, experts usually try to manually retrace the fragmentation events leading to the fragmentation pattern measured in the mass spectrometer. We model these fragmentation cascades using *fragmentation trees*. In a fragmentation tree, nodes are annotated with the molecular formulas of fragment ions, and edges with fragmentation events, that is, neutral or radical losses. The root of the fragmentation tree is labeled with the molecular formula of the molecular ion.

We present a novel computational method for the *de novo* interpretation of EI fragmentation data, based on fragmentation tree construction [61]. Besides a list of common neutral losses, our method does not use any chemical expert knowledge, and it is fully independent of databases. On the other hand, our method *does* require high mass accuracy of the measurements.

In this chapter, we describe the graph theoretical model for fragmentation tree computation, including graph construction, scoring, and finding the best scoring tree. We focus primarily on the handling of hard ionization issues, mainly the identification of low-abundance molecular ions (see Section 2.4.1). Given the EI fragmentation data of an unknown small molecule, our method tries to pick the molecular ion peak using hypothesis-driven evaluation of the data. Second, a molecular formula is derived for the hypothetical molecular ion peak. For optimal performance, molecular ion peak and formula identification are performed simultaneously. Third, we compute a fragmentation tree that offers a hypothetical interpretation of the experimental data.

We apply our method to two different datasets to evaluate the identification of molecular ions and molecular formulas. Further, we discuss the quality of the constructed fragmentation trees on selected examples taken from the literature. Even though we do not claim the pathways in the trees to be “true” fragmentation processes, we show that they agree in their general information very well with expert knowledge of EI fragmentation patterns.

4.1 Computing Fragmentation Trees

Fragmentation trees were introduced for the analysis of tandem MS data [7, 112] with known molecular ion mass. However, EI results in a mass spectrum not necessarily containing the molecular ion peak. For ease of presentation, we will at first assume that the molecular formula of the compound is known. Later, in Section 4.1.4, we will describe how to overcome the problems resulting from hard ionization.

4.1.1 General Fragmentation Model

Unlike proteins and glycans, metabolites can fragment at almost any chemical bond, and the fragmentation process is not completely understood and therefore difficult to

predict [164]. We account for missing comprehension by allowing arbitrary fragmentation. For this, the EI fragmentation spectrum is transformed to a fragmentation graph, modeling all possible fragmentation steps.

First, we compute candidate molecular formulas for each peak of the fragmentation spectrum. That is, we compute all molecular formulas that are within the mass accuracy of the instrument, and that are sub-formulas of the compound molecular formula. These candidate molecular formulas are called *decompositions* of the peak. For this, a set of potential chemical elements, which is called the alphabet in the following, must be provided to the method. A common choice is CHNOPS, that is, the six elements most abundant in metabolites, namely carbon (C), hydrogen (H), nitrogen (N), oxygen (O), phosphorus (P), and sulfur (S). We discard formulas that do not obey Senior’s third theorem [134].

We use these molecular formulas to label the nodes in the fragmentation graph. The nodes are colored, such that all explanations of the same peak receive the same color. Two nodes are connected by an edge (corresponding to a loss) if the second molecular formula is a sub-formula of the first. Since the “sub-formula” relation is transitive, the constructed graph is also transitive.

The resulting graph is a directed acyclic graph, since fragments can only lose, never gain, weight. The fragmentation graph contains all possible fragmentation trees as subgraphs.

4.1.2 Weighting the Fragmentation Graph

We weight nodes and edges of the fragmentation graph using log odds and log likelihoods. This enables a statistical interpretation of the outcome (i.e., maximum likelihood) [112]. The scoring scheme is similar to the one introduced by Böcker and Rasche [7].

Weighting nodes. We weight nodes (i.e. fragments) using mass deviation and peak intensity. We use log odds to differentiate between the model (the peak is truly a fragment with the proposed molecular formula) and the background (the peak is noise). We use the mass deviation between the measured peak and the molecular formula to assess whether the molecular formula is true. Mass deviations are assumed to be normally distributed [66, 170]. We evaluate the logarithmized Gaussian probability density function with standard deviation $1/3$ of the measuring mass error to score the node. We use the maximum of a relative error α and an absolute error β . We find that more intense peaks (in our system, higher than ca. 5000 counts) are spread more broadly due to their increased FWHM (full width at half maximum) in continuum mode. Therefore, we slightly increase the allowed error with increasing peak intensity by linearly interpolating between (α_0, β_0) at relative intensity 0% and (α_1, β_1) at relative intensity 100%. A heteroatom is any atom that is not carbon or hydrogen. Very unlikely molecular formulas with a ratio of carbon atoms to hydrogen atoms and heteroatoms above 3, are penalized by a constant $\log(\omega), \omega \ll 1$.

To identify noise peaks we use the peak intensity. We weight the peak intensities assuming noise peak intensities to be Pareto distributed. The peak intensity multiplied by a constant λ is logarithmized and added to the peak score.

The score of a node is pulled up to all incoming edges of the node. Constructing a solely edge-weighted graph simplifies further calculations.

Weighting edges. We weight edges (i.e. fragmentation reactions) according to their plausibility as real fragmentation steps. For EI, fragmentation mechanisms are well

understood. This provides us with a list of neutral and radical losses that appear more or less frequently when analyzing organic and biological compounds [50]. We classify these losses by their frequency of occurrence (see Table 4.1).

Our scoring system for losses involves parameters $\gamma_1 > \dots > \gamma_4 > 1$, ρ_1 , ρ_2 , and ϵ . We reward the occurrence of a type x loss by adding $\log(\gamma_x)$ with $\gamma_1 > \dots > \gamma_4 > 1$. We allow a combination of two losses of type x_1 and x_2 with a combined score $\log(\gamma_{\max\{x_1, x_2\}})$. Combinations may represent groups detaching together or the loss of an intermediate peak. Losses not contained in one of the groups are slightly penalized by adding $\log(\rho_1)$, $\rho_1 < 1$, and even more if they do not obey Senior’s third theorem [134], by adding $\log(\rho_2)$, $\rho_2 < 1$. Unlikely losses consisting purely of carbon or nitrogen are penalized by adding $\log(\epsilon)$, $\epsilon \ll 1$.

To avoid star-like fragmentation trees where all fragment ions branch directly from the molecular ion, we penalize large losses by $\log(1 - \frac{\text{mass(loss)}}{\text{highest peak mass}})$. Due to this score, fragments are inserted rather too deep than too high (see Sections 4.1.3 and 4.2.3 and Figure 4.1).

In organic compounds there is typically a carbon backbone complemented by some heteroatoms, such as nitrogen or oxygen. The heteroatom-to-carbon ratio is an indicator whether a molecular formula is possible. We use the density function of the normal distribution to score the heteroatom-to-carbon ratio of the decompositions, with mean 0.59 and SD 0.56 for calculations for halogen-free molecules and mean 0.53 and SD 0.52 for molecules containing halogens. In case no carbon is contained in the molecular formula of a fragment ion, we set the number of carbons to 0.8, to avoid division by zero. For further information on the scoring parameters, see Böcker and Rasche [7] and Rasche *et al.* [112].

4.1.3 Calculating Fragmentation Trees

To find the best explanation of the observed data we consider every subtree of the fragmentation graph that is rooted in the node of the molecular formula of the compound, as a hypothetical fragmentation tree. Considering *trees*, every fragment is explained by a unique fragmentation pathway; considering only *colorful* trees, every peak is explained by a single fragment (to avoid peak double-counting). Several fragments resulting in a single peak is an extremely rare event in practice. The colorful subtree with *maximum sum of edge weights* is the explanation of the observed fragments that fits best with the given conditions.

By demanding that each fragment in the fragmentation spectrum be generated by a single fragmentation pathway we slightly oversimplify the problem. Our optimization algorithm will choose one pathway for each fragment that is hopefully the most likely fragmentation reaction creating this fragment. There are two exceptions to this reasoning:

1. In the resulting fragmentation tree, assume that some fragment f_3 is cleaved from f_2 , which is in turn cleaved from f_1 . Solely from the EI fragmentation pattern and without additional structural information, it cannot be ruled out that fragment f_3 is in truth cleaved directly from f_1 . However, both interpretations are implicitly encoded in the fragmentation tree: the fragmentation may occur from the fragment’s direct parent in the tree or from any of its parents (see Figure 4.1(a)). An example are losses NO and O, where both NO and the combined loss NO_2 are characterized as being frequent (see Table 4.1 and Figure 4.8 (bottom)).

Table 4.1: List of neutral and radical losses used for scoring fragmentation reactions [50]. The losses are sorted by integer mass and their probability of occurrence in a GC-EI MS spectrum. Losses in group 1 are very common and thus score high, whereas losses in group 4 are relatively uncommon and thus score comparatively low.

integer mass	frequency of occurrence			
	group 1	group 2	group 3	group 4
1			H	
2				H ₂
3			H ₃	
15	CH ₃			
16		H ₂ N		O
17	H ₃ N	OH		
18	H ₂ O			
19		F		H ₃ O
20	HF			
26	C ₂ H ₂			CN
27	HCN			C ₂ H ₃
28	CO, C ₂ H ₄			N ₂
29	C ₂ H ₅	CHO		
30	NO			CH ₂ O
31	CH ₃ O			
32			CH ₄ O	S
33		CH ₅ O		HS
34	H ₂ S			
35		Cl		
36	HCl			
41	C ₃ H ₅		C ₂ H ₃ N	
42		C ₃ H ₆	C ₂ H ₂ O	
43	C ₃ H ₇ , C ₂ H ₃ O			
44	CO ₂	C ₂ H ₄ O		
45	C ₂ H ₅ O	CHO ₂ , C ₂ H ₇ N		
46	C ₂ H ₆ O	NO ₂		
48				SO
55			C ₄ H ₇	
56		C ₄ H ₈	C ₂ O ₂	
59	C ₂ H ₃ O ₂			
60	C ₂ H ₄ O ₂			
72			C ₂ O ₃	
73		C ₃ H ₅ O ₂		
77	C ₆ H ₅			
78	Br			
89	OTms			
91	C ₇ H ₇			
126		I		
181	Pfb			
197	PfbO			
198	PfbOH			

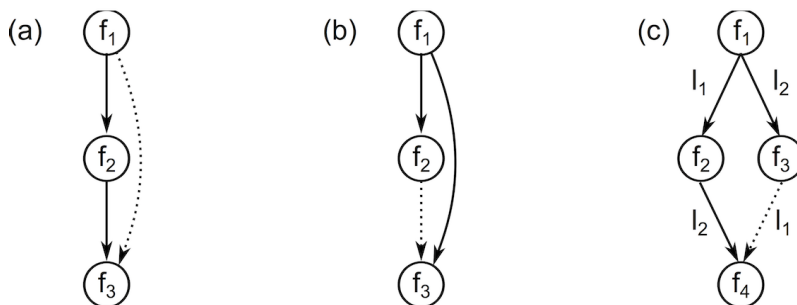


Figure 4.1: Computing trees does not allow the explanation of a fragment by more than one fragmentation pathway. Our optimization algorithm will choose the most likely pathway to compute a fragmentation tree (solid edges). Other explanations (dotted edges) are lost. However, in case (a) and case (c) the alternative pathway is implicitly encoded in the fragmentation tree. (a) In the fragmentation tree, fragment f_3 is cleaved from f_2 , which is in turn cleaved from f_1 . It cannot be ruled out that fragment f_3 is in truth cleaved directly from f_1 (dotted edge). Both interpretations are implicitly encoded in the fragmentation tree. We evaluate these edges as *correct*. (b) In the fragmentation tree, fragment f_3 is cleaved directly from f_1 , while in truth it is cleaved from f_2 (dotted edge), which is in turn cleaved from f_1 . We evaluate these edges as inserted *too high*. (c) *Parallelogram*: Fragment f_4 is cleaved from f_2 by losing l_2 , which is in turn cleaved from f_1 by losing l_1 . In truth fragment f_4 is cleaved by losing l_2 first and l_1 afterwards (dotted edge). Both interpretations are implicitly encoded in the fragmentation tree. We evaluate these edges as *correct*.

2. In the resulting fragmentation tree, assume that some fragment f_2 is cleaved from a fragment f_1 by losing l_1 and another fragment f_3 is cleaved from f_1 by losing l_2 . Further, another fragment f_4 is cleaved from f_2 by losing l_2 . Solely from the data, it cannot be ruled out that fragment f_4 is in truth cleaved from f_3 by losing l_1 . Again, both interpretations are implicitly encoded in the fragmentation tree: the fragmentation may occur by losing l_1 first and l_2 afterwards, or *in reverse order* (see Figure 4.1(c)). An example is the pair of losses H_2O and C_2H_4 which are both characterized as being frequent (see Table 4.1) and are cleaved successively from the same fragment in Figure 4.8 (top), where both intermediate fragments are detected. In the following, we call this configuration a *parallelogram*.

Calculating fragmentation trees under the described conditions is formalized as the MAXIMUM COLORFUL SUBTREE problem [7].

MAXIMUM COLORFUL SUBTREE problem.

Given a node-colored DAG $G = (V, E)$ with colors \mathcal{C} and weights $w : E \rightarrow \mathbb{R}$, find the colorful subtree $T = (V_T, E_T)$ of G of maximum weight $w(T) := \sum_{e \in E_T} w(e)$.

The MAXIMUM COLORFUL SUBTREE problem is NP-hard [32] as well as APX-hard [25] even on binary trees. Furthermore, on general trees it has no constant factor approximation [25, 136]. For the analysis of multiple-stage mass spectrometry data (see Section 2.4.2), Scheubert *et al.* [125] present the related COLORFUL SUBTREE CLOSURE problem.

Several exact and heuristic algorithms have been developed for the MAXIMUM COLORFUL SUBTREE problem [7, 114]. Here, we focus on two algorithms that guarantee to find the optimal solution and are also swift in practice. The problem can be solved exactly using dynamic programming (DP) over nodes and color subsets [27]. This yields a

fixed-parameter algorithm as the number of colors k , representing the number of peaks in the input spectra, restricts the exponential growth. We can compute a maximum colorful tree in $O(3^k k |E|)$ time and $O(2^k |V|)$ space. Running time can be improved using subset convolutions and the Möbius transform [5], but this is of theoretical interest only. Due to the exponential running time and space, exact calculations are limited to $k \leq k'$ colors for some moderate k' (see Section 4.1.4). In addition to the DP algorithm, an Integer Linear Program (ILP) for the MAXIMUM COLOURFUL SUBTREE problem has been constructed [114]. ILPs have proven useful in providing quick exact solutions to NP-hard problems.

4.1.4 Handling Hard Ionization Issues

For the description of the fragmentation model in Section 4.1.1, we assumed the molecular formula of the compound to be known. Based on the technical setup, for tandem and multiple MS fragmentation data, the molecular ion peak is commonly recorded in an MS¹ spectrum. Thus, for the construction of tandem and multiple MS fragmentation trees, at least the molecular ion peak is known [112, 124] (see Sections 2.4.2 and 3.6). In contrast, EI is a hard ionization technique that simultaneously ionizes and fragments the molecule and results in missing or low intensity molecular ion peaks [78, 79]. Therefore, we do not know the true molecular formula of the compound, the mass of the molecular ion, or even whether the molecular ion peak is contained in the spectrum at all. In our analysis we assume that the molecular ion peak is present but possibly of very low intensity.

Because the molecular formula of the compound is unknown, molecular formulas explaining each peak cannot be restricted to sub-molecular formulas as proposed in Section 4.1.1. This drastically increases the complexity of the problem. This is especially pronounced for compounds above 500 Da, or fragmentation spectra with many peaks, and it is aggravated if we consider elements besides CHNOPS. In addition, all nodes in the graph are possible roots of the fragmentation tree. Let us consider the relatively small compound naphthalene (128 Da; see Table A.2 in the appendix). The EI spectrum shows 52 peaks that have at least one decomposition. Over all compounds in our dataset, this is far below the average number of peaks with decomposition. For this compound, we would have to consider roughly 10^{100} (one googol) fragmentation trees that are simultaneously contained in the fragmentation graph – a number much larger than the number of atoms in the observable universe. For compounds with more than hundred peaks with at least one decomposition, this number increases far beyond 10^{260} trees. Among all these potential fragmentation trees we want to find the single one with optimal score, a computationally very demanding task. To this end, we adopted a two-step approach (see Figure 4.2).

Step 1 – Finding the molecular ion peak and molecular formula. We simultaneously solve the problems of finding the molecular ion peak *and* the corresponding molecular formula. We compute a fragmentation tree using only a set of peaks that appear to be most relevant for a compound. These peaks are selected using three different criteria. Obviously, intense peaks seem to be relevant so we use the k_1 most intense peaks, for some parameter k_1 . However, EI fragmentation often results in low abundances of larger fragments. To include peaks with higher m/z values, we use a score for combining peak intensities with m/z values of the peaks, namely

$$m/z \cdot \ln(100 \cdot \text{int}_{rel}) \quad (4.1)$$

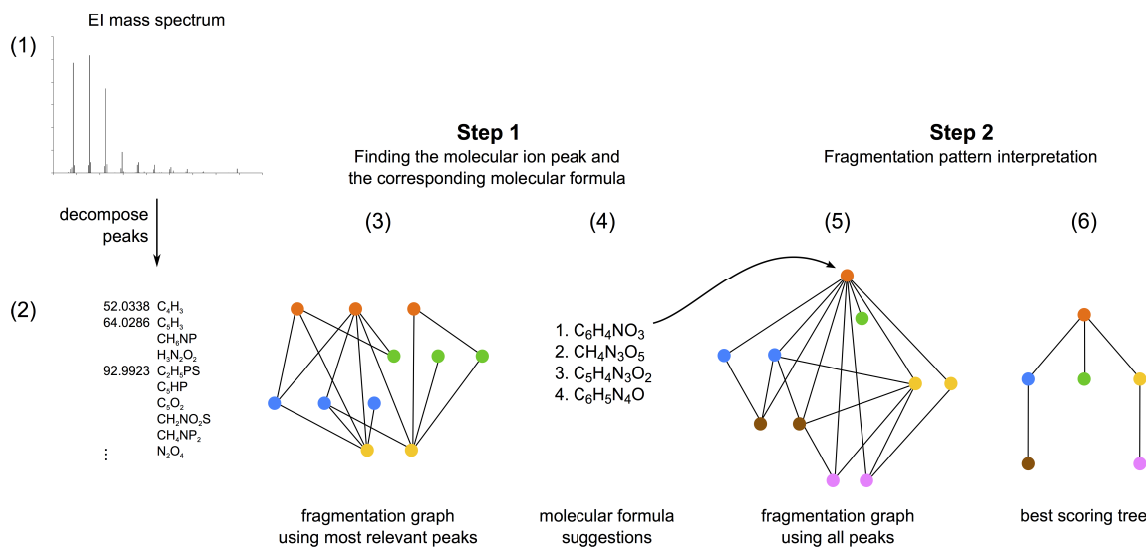


Figure 4.2: Algorithmic workflow. (1) As input we use a EI mass spectrum with high mass accuracy. Isotopic peaks are removed (details omitted). (2) Candidate molecular formulas are computed for each peak using the mass accuracy of the instrument. *Step 1 - Finding the molecular ion peak and the corresponding molecular formula:* (3) A fragmentation graph is constructed using only a set of peaks that appear to be most relevant for a compound. Each of these peaks is considered a potential molecular ion peak. Explanations of the same peak receive the same color. (4) Molecular formulas of the different potential molecular ion peaks are ranked according to the score of the fragmentation trees rooted in this molecular formula. *Step 2 - Fragmentation pattern interpretation:* (5) A fragmentation graph rooted in the correct molecular formula of the compound is constructed using all peaks. (6) A hypothetical fragmentation tree is computed that best explains the observed data.

where m/z is the m/z of each peak, and int_{rel} is its relative intensity. Here, we add the k_2 best scoring peaks according to (4.1). To increase chances of including the correct molecular ion peak in our computation, we also use the k_3 best scoring peaks in the *upper m/z range*, which we define as the m/z region from $0.9\tilde{M}$ to \tilde{M} where \tilde{M} is the highest m/z of a peak detected in the spectrum. In this step, fragmentation trees are computed using Dynamic Programming (DP) [7], as we have limited the number of colors to some moderate $k \leq k_1 + k_2 + k_3$ (see Section 4.1.3). The advantage of DP is, that we have to fill the DP table only once to get the results for all candidate molecular formulas explaining the potential molecular ion peaks. Therefore, we can consider each of the selected peaks as the potential molecular ion peak since running time only depends on the number of selected peaks and not on the number of potential molecular ion peaks.

Molecular formulas of the different potential molecular ion peaks have to observe two characteristics. The molecular ion must be a radical cation and therefore must have an odd number of electrons. Further, an odd nominal integer molecular mass has to imply an odd number of nitrogens. These restrictions are not used for fragment formulas. Molecular formulas of the different potential molecular ion peaks are then ranked according to the score of the fragmentation trees rooted in this molecular formula.

Step 2 – Fragmentation pattern interpretation. In the second step, we compute a fragmentation tree for the complete spectrum assuming that we know the correct molecular

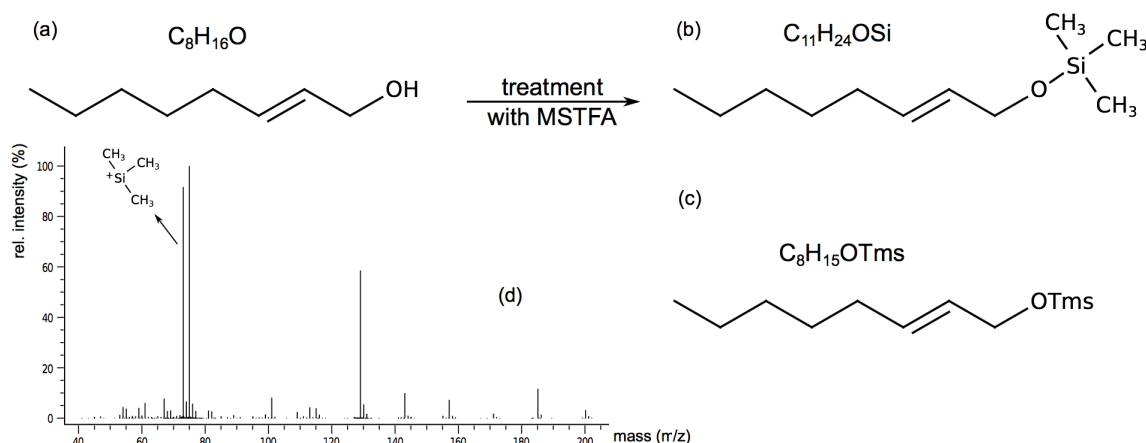


Figure 4.3: Derivatization of compound (E)-oct-2-en-1-ol (CAS 18409-17-1) (a). The compound is treated with MSTFA. The active hydrogen is replaced by a trimethylsilyl (TMS) group $[-\text{Si}(\text{CH}_3)_3]$. (b) The resulting derivate (E)-trimethyl(oct-2-enyloxy)silane is more volatile, less polar, and more thermally stable than its precursor. (c) We introduce the artificial element Tms to avoid the incorporation of silicon into the molecular formula that is not part of the TMS group. (d) The EI spectrum of (E)-trimethyl(oct-2-enyloxy)silane contains the typical signal after silylation, that is, the stable ion with m/z 73.047 ($\text{Si}(\text{CH}_3)_3^+$).

ion and molecular formula of the compound. The resulting fragmentation tree is rooted in this molecular formula. Although we can restrict molecular formulas of fragments to sub-formulas of the compound molecular formula, we have to consider a huge number of trees. For large compounds, such as octacosane (394 Da), we again have to consider roughly 10^{100} different trees, even if we assume that we know the molecular formula. In this step, fragmentation trees are computed using Integer Linear Programming [114] to find the single fragmentation tree with maximum score. Evaluations by Rauf *et al.* [114] clearly indicate that heuristic algorithms lead to fragmentation trees of much inferior quality.

4.1.5 Dealing with Derivatizations

In a typical metabolomics experiment the sample is treated with MSTFA first and PFBHA afterwards [159] (see Section 2.4.1). Typical signal after silylation is the stable ion with m/z 73.047 ($\text{Si}(\text{CH}_3)_3^+$), referred to as TMS (see Figure 4.3). A frequently occurring m/z after derivatization with PFBHA to form the respective oxime is m/z 181.007 ($\text{C}_7\text{H}_2\text{F}_5^+$), referred to as PFB. We use these typical m/z values to identify derivatized compounds.

We introduce the artificial elements Tms (mass 73.047 Da for molecular formula SiC_3H_9) and Pfb (mass 181.008 Da for molecular formula $\text{C}_7\text{H}_2\text{F}_5$) to our alphabet modeling the groups originating from derivatization. This is done to avoid the incorporation of fluorine or silicon into any molecular formula when it is not part of TMS or PFB. Other elements can be flexibly adapted on demand.

4.2 Evaluation of Fragmentation Tree Quality

In this section, we evaluate the identification of molecular ions and molecular formulas of unknown molecules using fragmentation trees and further show that fragmentation trees are a viable explanation of the spectrum. We compare the calculated trees against expert

knowledge and annotated pathways from the literature and evaluate our method against two approaches which have found wide acceptance in the community.

4.2.1 Datasets and Parameter Settings

We use two datasets for the evaluation of our method (see Table 4.2). To evaluate the capability of fragmentation trees to reconstruct EI fragmentation processes we extract annotated fragmentation pathways for 22 compounds from different compound classes and simulate spectra from the pathways (*simulated data*) [57]. To further evaluate whether the method can be applied to automatically analyze authentic fragmentation mass spectra and support the explorative character of metabolomics studies, we use measure mass spectra for a set of 50 compounds using GC/TOF-MS (*measured data*) [61].

Simulated data. For this dataset, we use 22 compounds with fragmentation pathways that are well annotated in the literature [1, 19, 45, 99, 110, 153]. The spectra are given in the publications in various formats, e.g., plots or tables containing nominal masses and relative intensities. Measured spectra (not to mention high mass accuracy spectra) are not available to us. Thus, we simulate spectra from the pathways. From the molecular formulas in the fragmentation pathway, we compute exact peak masses, and simulate “measured” spectra by adding a normally distributed error of 10 ppm to the mass of the fragment formula (ignoring ionization). Peak intensities of the fragments are taken from the literature. In case the intensities are not written down in the respective publication they are estimated from the plotted spectrum. In addition, we add 70% noise peaks with uniformly distributed masses smaller than the mass of the molecular ion, and Pareto distributed intensities. For the full list of compounds, see Table A.1 in the appendix.

Measured data. For this dataset, we measure mass spectra for a set of 50 compounds from a wide array of compound classes ranging from structurally simple compounds, such as alcohols, to more complex compounds, such as steroids. Some of the compounds were treated with PFBHA (pentafluorobenzylhydroxylamine) or MSTFA (N-methyl-N-trimethylsilyl-trifluoroacetamide) for derivatization. The compounds are analyzed on a GCT-Premier (Waters-Micromass UK) time-of-flight mass selective detector coupled to an Agilent 6890N gas chromatograph. The instrument was calibrated using heptacosafuorotributylamine (heptacos) ion signals. For all of the compounds, the molecular ion peak is present in the spectrum. For 13 compounds, the relative intensity of the molecular ion peak is below 5% (*challenging compounds*). The expert estimate of measurement accuracy is 10 ppm. For the full list of compounds, see Table A.2 in the appendix.

In the presented analysis we use the following parameters for both datasets: We choose $\lambda = 0.1$, $\alpha_0 = 7$ ppm and $\beta_0 = 5$ mDa (at low intensity), and $\alpha_1 = 25$ ppm and $\beta_1 = 30$ mDa (at full intensity), and $\omega = 5 \cdot 10^{-5}$ for node scoring. Further, we choose $\gamma_4 = 5$, $\gamma_3 = 10$, $\gamma_2 = 50$, $\gamma_1 = 100$, $\rho_1 = 0.1$, $\rho_2 = 0.25$ and $\epsilon = 10^{-4}$ for edge scoring.

4.2.2 Identification of Molecular Ion Peaks and Molecular Formulas

Fragmentation trees enable the identification of the molecular ion and the molecular formula of a metabolite if the molecular ion is present in the spectrum. EI is a

Table 4.2: The two datasets used in this study. ^aNumber of compounds in the datasets; ^balphabet of potential elements provided to the method for molecular ion and molecular formula identification.

	# ^a	instrument	alphabet ^b	mass range	average
simulated data [57]	22	simulated		41.0 – 518.2 Da	187.6 Da
standard compounds	20		CHNOPS	41.0 – 518.2 Da	170.7 Da
chlorinated compounds	2		CHNOPSCl	277.0 – 399.2 Da	338.1 Da
measured data [61]	50	GC/TOF-MS		82.1 – 412.4 Da	212.9 Da
standard compounds	35		CHNOPS	82.1 – 412.4 Da	202.0 Da
chlorinated compounds	5		CHNOPSCl	128.0 – 191.0 Da	153.2 Da
derivatized compounds	10		CHNOPSTmsPfb	158.1 – 376.3 Da	280.9 Da

hard ionization technique resulting in missing molecular ion peaks in about 30 % of the spectra [89].

To evaluate whether molecular ion peaks of low intensity are a common phenomenon, we inspected EI mass spectra in the Golm Metabolome Database (GMD) [83]. This freely available database contains measurements with unit mass accuracy. We use these measurements to evaluate the presence and intensity of the molecular ion peak. There are 1426 compound entries in the GMD with molecular mass information. Of these spectra, 42 % do not contain the molecular ion peak and hence cannot be analyzed by the method presented here. This is higher than the above estimate of 30 % [89]. For the remaining 828 spectra, about 65 % show an ion peak that has less than 5 % relative intensity, whereas in our dataset only 13 (26 %) show a molecular ion peak with intensity this low. Our method tries to pick the molecular ion peak even for these rather difficult spectra.

To identify the molecular ion peak and its formula, an alphabet of potential elements must be provided to the method. As running time increases with increasing size of the alphabet, we identify the molecular formulas under different conditions for the two datasets (see Table 4.2). For all compounds, we use the six elements most abundant in metabolites, namely carbon (C), hydrogen (H), nitrogen (N), oxygen (O), phosphorus (P), and sulfur (S) [65]. When analyzing chlorinated compounds (two compounds in the *simulated* dataset and five compounds in the *measured* dataset), we also add chlorine (Cl) to the alphabet. Information on whether or not a compound contains chlorine can be usually obtained from isotope pattern analysis, as the nucleon numbers of the stable chlorine isotopes are separated by 2 (i.e. ³⁵Cl and ³⁷Cl) and show therefore a characteristic isotope pattern. For all compounds in the *measured* dataset that were potentially derivatized during sample preparation, we add the artificial elements Tms and Pfb to our alphabet. We then test whether the characteristic peaks at m/z 73.047 and m/z 181.007 are present in the spectrum; if so, we force all molecular formulas of the potential molecular ion peaks to contain the corresponding derivatization at least once, and do not use the other derivatization as an artificial element. In cases where no characteristic peak is present we add both artificial elements to the alphabet as, in practice, we would not know the derivatization of the compound.

For the *simulated data*, computing the molecular ion peak and molecular formula requires an average of 4.6 s for each compound. This time includes peak decomposition and graph construction. We discard peaks with no decomposition. We then choose the subset of peaks that appear to be most relevant for the compound as described in Section 4.1.4. We choose $k_1 = 10$ and $k_2 = k_3 = 5$, resulting in at most 20 peaks if the sets are not overlapping. Our method correctly detects the molecular ion peak for all compounds. This is not surprising

Table 4.3: Results for the identification of the molecular ion peaks and molecular formulas for the *measured data*. Total numbers and percentage of compounds with correct suggestion in the top 1, 3, and 5 suggestions. Results are shown for *all* 50 compounds and in particular for the 13 *challenging* compounds with molecular ion peak below 5 % relative intensity.

compounds:	molecular ion		molecular formula	
	all	challenging	all	challenging
top 1	44 (88 %)	12 (92 %)	39 (78 %)	12 (92 %)
top 3	50 (100 %)	13 (100 %)	44 (88 %)	12 (92 %)
top 5	50 (100 %)	13 (100 %)	49 (98 %)	13 (100 %)

as in each spectrum, the molecular ion peak has the highest m/z value since we have simulated noise peaks with masses lower than the mass of the molecular ion only. For 17 of the 20 compounds (85 %), the highest scoring suggestion for both the molecular ion peak and its molecular formula is correct. For the remaining three compounds, the correct molecular formula is the second suggestion.

For the *measured data*, we choose a smaller subset of peaks, that is, $k_1 = 8$ and $k_2 = k_3 = 4$, resulting in at most 16 peaks if the sets are not overlapping. Here, computing the molecular ion peak and molecular formula requires an average of 2.5 s for each compound. For 44 of 50 compounds (88 %), our method correctly detects the molecular ion peak as the first suggestion (see Table 4.3). For 39 compounds (78 %), the highest scoring suggestion for both the molecular ion peak and its molecular formula is correct. For all but one compound, the correct molecular formula is in the top five molecular formulas suggested by our method. For this compound, namely anthracene (CAS 120-12-7), the spectrum has few significant peaks, that is, only five peaks with relative intensity above 5 %. Even for 12 of the 13 challenging spectra (92 %) with molecular ion peak below 5 % relative intensity, the molecular formula is correctly identified. For the other compound, the TMS derivate of arachidonic acid (CAS 506-32-1), the characteristic ion at m/z 73.047 is not contained in the spectrum. Therefore both artificial elements (Tms and Pfb) are added to the alphabet, making the identification much harder. Nevertheless, the correct molecular formula is identified at the fourth position. The full list of results can be found in Table A.3 in the appendix.

When analyzing halogenated compounds, we can use prior information on which halogens are present in a particular compound. Such information may be obtained from the isotope pattern analysis, but is not necessarily required for our method. We repeat the analysis for the five halogenated (chlorinated) compounds using CHNOPS and all halogens found in metabolites, namely fluorine (F), chlorine (Cl), bromine (Br) and iodine (I). For these five compounds, average running time slightly increases from 1.0 s to 1.3 s due to the increased size of the alphabet. For four compounds, the rank of the correct molecular ion and molecular formula remains the same. For the remaining compound, the rank of the correct molecular formula and molecular ion degrades from two to three.

Comparison to NIST MW Estimator [132]. We compare our method for molecular ion identification to Scott’s well established algorithm [132] for estimating the nominal mass of a compound (see Section 3.5.3). This algorithm uses pattern recognition to first classify the compound and then uses empirical linear corrections to estimate its nominal mass. This approach has found wide acceptance in the community as it is implemented in the NIST

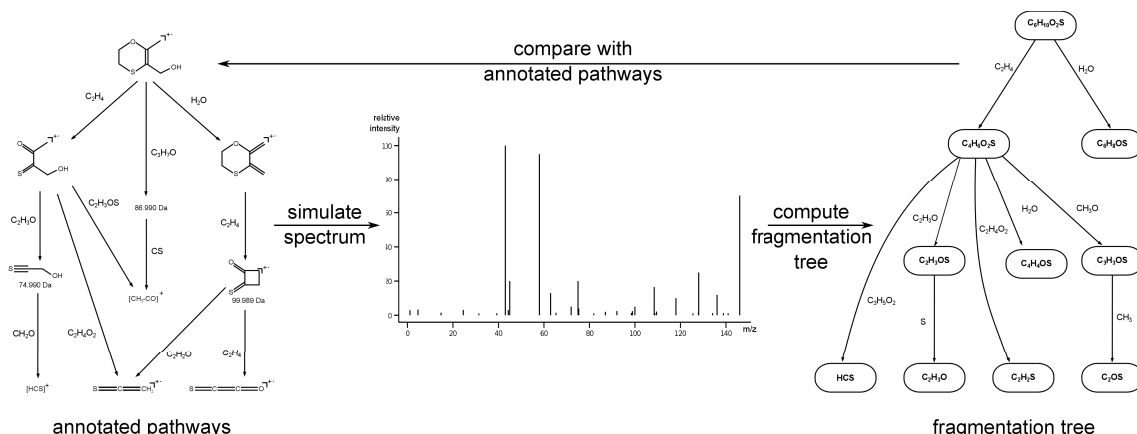


Figure 4.4: Evaluation scheme for the *simulated* dataset. For each compound in this dataset, a spectrum is simulated from the annotated fragmentation pathways: For each fragment formula, the exact peak mass is calculated and a measurement error is added. Peak intensities are taken from the literature. Additional noise peaks are added to the spectrum. From the simulated spectrum, a fragmentation tree is computed using our method. The computed tree is compared to the annotated pathway by evaluating whether the fragments are inserted *correctly*, *too deep*, *in reverse order*, or *too high*. Evaluation of the depicted compound (5,6-hydro-3-hydroxymethyl-2-methyl-1,4-oxathiine) is shown in Figure 4.5.

software. We use NIST MS Search Software version 2.0f (demo version) for estimating the nominal mass of all compounds in the *measured* dataset. The full list of results can be found in Table A.3 in the appendix. Scott’s algorithm estimates the correct nominal mass for 45 compounds. Our method detects the correct molecular ion for 44 compounds. Except for one compound (CAS 110-83-8) the results of the two methods complement each other: that is, either Scott’s algorithm or our algorithm (or both) are able to infer the correct nominal mass. Recall that Scott’s algorithm does not infer exact masses or molecular formulas.

4.2.3 Evaluation against Annotated Fragmentation Pathways

The *simulated data* is based on fragmentation pathways extracted from the literature. We use this dataset to evaluate the quality of computed fragmentation trees by comparing them to the annotated fragmentation patterns (see Figure 4.4). We compute a hypothetical fragmentation tree for every compound, assuming that we know the correct molecular ion and molecular formula of the compound. In this step, all peaks of the spectrum are used for computation. Computation, including decomposition and graph construction, requires 1.5s on average and a maximum of 18.5s for the largest compound, namely gossypol. For this compound with a mass of 518.2 Da, decomposition of all peaks requires 17.9s (97% of the total running time).

The fragmentation trees annotate 284 peaks in total (see Table 4.4). Only seven of these explanations (2.5%) are false positives, that is, explanations of artificially generated noise peaks as fragments. The remaining 277 peaks are annotated with the correct fragment formula. From all 296 fragments described in the pathways from literature, 19 (6.4%) could not be explained. There are different reasons for a peak not being explained in the tree. For some peaks, the mass deviation between the measured peak mass and the

Table 4.4: Peak explanations in the annotated pathways compared to the computed fragmentation trees for the *simulated data*. ¹Percent of the explanations in the annotated pathways. ²Percent of the explanations in the computed fragmentation trees.

	pathway total	tree total	correct	missing	additional
peak explanations	296	284	277	19	7
percentage			93.6 % ¹	6.4 % ¹	2.5 % ²

exact mass is too high. This effect becomes stronger for smaller peaks, since the mass deviation penalty is dependent of the peak intensity (see Section 4.1.2). For other peaks, the fragmentation step resulting in this fragment gets a bad score. For example, the loss C_2H_2N that was annotated in the literature as a first fragmentation step for three of the five alkyl isocyanides is not included in the list of common losses for EI fragmentation and is not even a combination of these (see Table 4.1). Therefore, the fragments resulting from this step could not be identified. Nevertheless, the method is capable of identifying losses that are very specific to a single compound or compound class and therefore not listed as a common loss (see Section 4.2.4).

Further, we compare edges from the fragmentation tree to those in the annotated pathways. Matching losses are assigned as *correct*. In some cases, consecutive edges of the fragmentation tree can be combined to give the molecular formula of a single fragmentation step in the annotated fragmentation pathways (see Figure 4.1(a)). In some other cases two consecutive losses in the fragmentation tree are described in reverse order in the annotated fragmentation pathways (see Figure 4.1(c)). We evaluate those fragments that are inserted *too deep* or in *reverse order* in the fragmentation trees as *correct*, since without a given structural formula and solely from the EI fragmentation data, the correct case cannot be distinguished from our method’s suggestion. If the fragmentation step in the resulting fragmentation tree is explained by several consecutive steps in the annotated pathway, the fragment is inserted *too high* (see Figure 4.1(b)). If the fragment is inserted into a completely different pathway the edge is assigned as *wrong*.

For 5,6-hydro-3-hydroxymethyl-2-methyl-1,4-oxathiine [99], we now describe in more detail how we evaluate the edges of the fragmentation tree (see Figure 4.5). We choose this compound as a worst-case example to visualize all the things that can go wrong. We use the following notation throughout the evaluation: (146-118) is an edge connecting the nodes at m/z 146 and m/z 118. The loss of ethene from the molecular ion (146-118) followed by a loss of C_2H_3O (118-75) as well as a loss of $C_2H_4O_2$ (118-58) are annotated as *correct*, as they can be found in the annotated pathways. The water loss from the molecular ion (146-128) is also annotated in the literature. In the fragmentation tree ethene gets lost first and water afterwards (146-118-100), whereas in the annotated pathway these losses are cleaved in *reverse order*. Edges between nodes 118-75-43 can be combined to the expected loss of C_2H_3OS so the loss of sulfur is considered as *correct*. Pulling up the edges between nodes 146-118-87 results in a total loss of C_3H_7O , so the CH_3O loss was inserted *too deep* and is considered as *correct* by pull-up. Cleaving fragment 45 directly from 118 is considered as *too high*. Fragment 72 was cleaved by losing ethene from fragment 100 in the annotated pathway. Therefore the methyl loss (87-72) in the fragmentation tree is annotated as *wrong*.

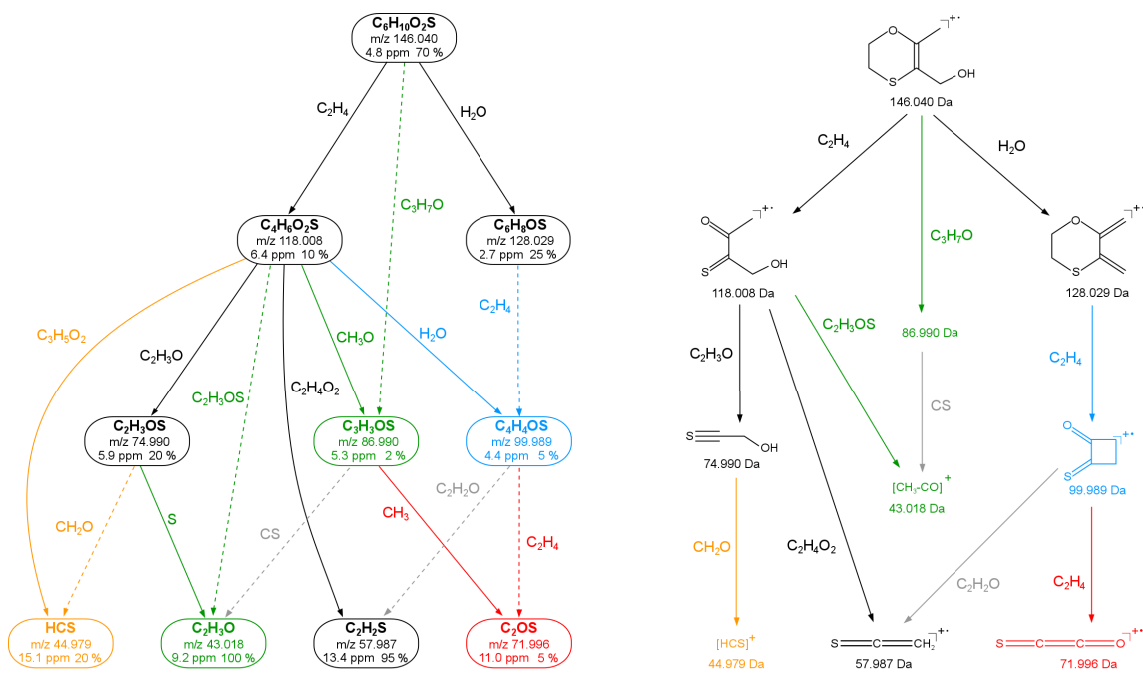


Figure 4.5: Computed fragmentation tree (solid edges) of 5,6-hydro-3-hydroxymethyl-2-methyl-1,4-oxathiine (left) compared to the annotated pathways [99] from the literature (right). This compound is a worst-case example to visualize all the things that can go wrong. Fragment formulas, m/z values, mass deviations and intensities are given in the nodes. All fragments are annotated with the correct molecular formula. Dashed edges in the tree are losses from the annotated pathways. Black edges in the fragmentation tree agree with the annotated pathways. Gray dashed edges are additional pathways that could not be computed since the tree property would have been violated. The blue fragment is actually cleaved in *reverse order* from the molecular ion. The green fragments are inserted *too deep*, and the orange fragment is inserted *too high* in the fragmentation tree. The red fragment is inserted into a completely different pathway. Note that mass errors of more than 10 ppm occur as we added the simulated mass error on the mass of the fragment formula (without considering ionization).

We use similar reasoning processes to evaluate all hypothetical fragmentation trees (see Figure 4.6 for two further examples and Table 4.5 for an overview). For 5 of the 277 correct peak explanations, the fragmentation process leading to this fragment is not given in the literature. From the remaining 272 losses in our data set, 214 losses (78.7%) are assigned as *correct*. From these, 31 fragments (11.4%) are inserted *too deep* and 8 fragments (2.9%) are actually cleaved in *reverse order*. Further, we find that 19 fragments (7.0%) are inserted *too high* and 39 edges (14.3%) are annotated as *wrong*. We stress that, unlike

Table 4.5: Evaluation of the fragmentation events annotated in the fragmentation trees for the *simulated data*. For 5 of the 277 correct peak explanations, the fragmentation process leading to this fragment is not given in the literature.

	total	correct	correct, but <i>too deep</i>	correct, but <i>reverse order</i>	<i>too high</i>	wrong
losses	272	214	31	8	19	39
percentage		78.7 %	11.4 %	2.9 %	7.0 %	14.3 %

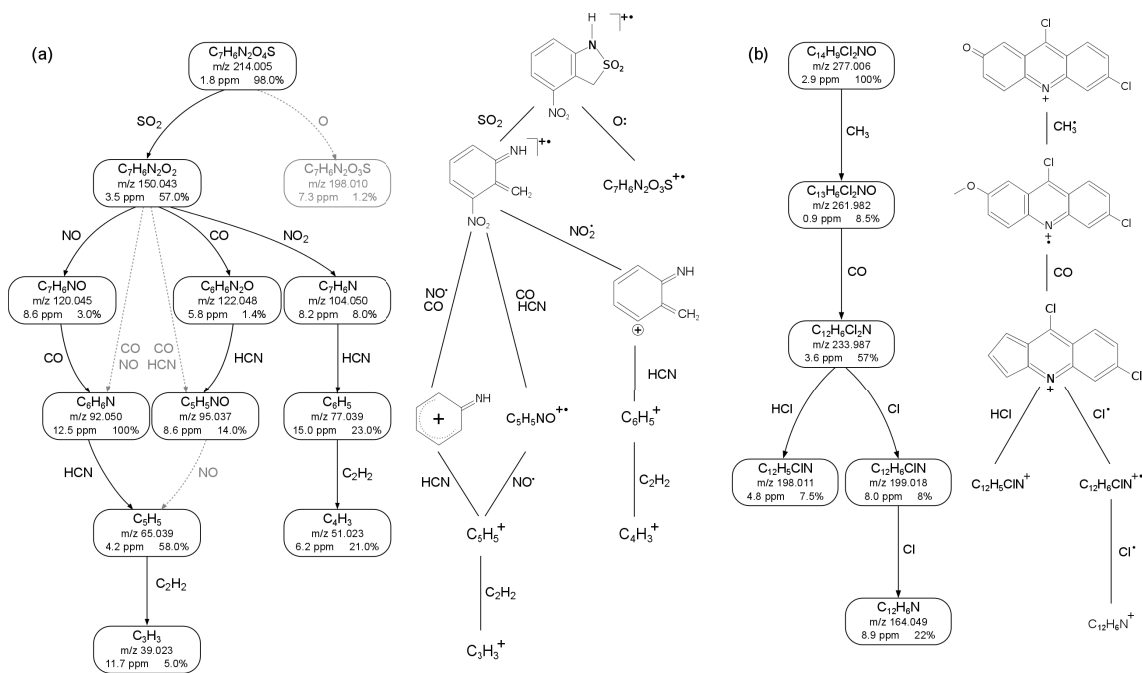


Figure 4.6: Fragmentation trees compared to annotated pathways from the literature. Fragment formulas, m/z values, mass deviations and intensities are given in the nodes. (a) Fragmentation tree (left) and annotated pathway (right) of a 2,1-benzisothiazoline 2,2-dioxide nitro derivative (compound 6 from [19]). The gray fragment is not explained in the fragmentation tree as it has very low intensity and results from a rather uncommon loss (see Table 4.1). Dashed edges in the tree are additional losses from the annotated pathways that cannot be explained by our method since the tree property would be violated. In the literature the edge (150-92) combines two fragmentation steps (150-120-92), since the m/z 120 peak is very small. In truth, it is very likely that this fragmentation always proceeds in two steps, but that the lifetime of the intermediate ions is too short [19]. The same applies to edge (150-95) combining the two fragmentation steps (150-122-95). (b) The fragmentation tree (left) and the annotated pathway (right) of 6,9-dichloro-2-methoxyacridine [1] match completely. Note that mass errors of more than 10 ppm occur since we added the simulated mass error on the mass of the fragment formula (without considering ionization).

for the annotation of the fragmentation processes in the literature, our method has no information about the molecular structure of the compounds.

Parallelograms. We evaluate the frequency of *parallelograms* in the annotated pathways from the literature (see Table 4.6). As mentioned above, these are configurations where it cannot be decided solely from the data whether a fragment results from cleaving loss l_1 first and l_2 afterwards or the other way round (see Figure 4.1), since both intermediate fragment ions are present in the spectrum. In total, we find 99 parallelograms, spanning all but three compounds. 29 of these are *closed*, that is, both fragmentation pathways are annotated. This is possible since pathways from the literature do not necessarily have to be trees. In contrast, our method has to choose one of these fragmentation pathways. For the remaining 70 parallelograms, exactly one of the two pathways is annotated. From these 70 parallelograms, our method selects the other (possibly wrong) pathway in only 8 (11.4 %) cases.

Table 4.6: Evaluation of the frequency of parallelograms in the annotated pathways from the literature (*simulated data*). A parallelogram is *closed* if both fragmentation pathways are annotated, and “*open*” otherwise. ³Percent of “open” parallelograms.

	total	<i>closed</i>	“ <i>open</i> ”	different in tree
parallelograms	99	29	70	8
percentage		29.3 %	70.7 %	11.4 % ³

4.2.4 Evaluation against Expert Knowledge

To evaluate the method’s capability to automate the analysis of authentic fragmentation mass spectra, we compute a hypothetical fragmentation tree for each compound in the *measured dataset*, assuming that we know the correct molecular ion and molecular formula of the compound. We use all peaks of the spectrum with relative intensity above 1 %, excluding peaks of isotopic distributions of the fragments (details omitted). Using all peaks is possible for this purpose because running time decreases significantly if the molecular formula of the compound is known. Computation, including decomposition and graph construction, required 0.7 s on average with a maximum of 5.6 s.

To evaluate our method, mass spectrometry experts experienced in the structural elucidation of natural products manually compared the fragmentation trees obtained by our program with fragmentations described in the literature. This evaluation was carried out by Rempt [115]. We use four published examples of different substance classes to validate the method (see Figures 4.7 and 4.8), and in addition, evaluate the fragmentation tree of one larger molecule (see Figure 4.9).

(2-Chloroethoxy) benzene (CAS 622-86-6) The first example is a molecule from the widely used mass spectrometry textbook “Interpretation of mass spectra” by McLafferty [96] (see Figure 4.7 (top)). When it fragments, this molecule loses chlorine ($M-Cl$)⁺ (156-121) and afterwards C_2H_4 (121-93). That the peak of ($M-CH_2Cl$)⁺ is higher than the ($M-Cl$)⁺ peak indicates a labile CH_2Cl group which can be cleaved in α -arrangement from the aromatic structure. This α -cleavage was found in the edge (156-121-107). A hydrogen rearrangement and the loss of C_2H_3Cl (156-121-94) is expected from this compound class because a stable phenol moiety is then formed. McLafferty does not explain further reactions, but according to our fragmentation analysis, the m/z 94 and m/z 93 fragments undergo further cleavage in typical fashion involving phenol-like keto-enol tautomeric reactions and CO loss (93-65) and (94-66). The occurrence of m/z 77 indicates monosubstituted phenolic moieties. We conclude that all edges in the fragmentation tree of (2-chloroethoxy) benzene are correct. In addition, we were able to confirm previously proposed fragmentation pathways [122].

Cyclohexene (CAS 110-83-8) This second example is a model for ring systems and alkenylic elements (see Figure 4.7 (bottom)). Fragmentation of this compound is also described by McLafferty [96] and it includes important reactions such as the α -cleavage and retro-Diels-Alder (rDA) reactions. In the fragmentation tree presented, the node with the highest score corresponds to heterolytic cleavage of a C-C bond followed by proton rearrangement and α -cleavage thus losing CH_3 (82-67). A rDA reaction produces the neutral molecule C_2H_4 and a very stable butadiene radical cation (C_4H_6)⁺ (82-54).

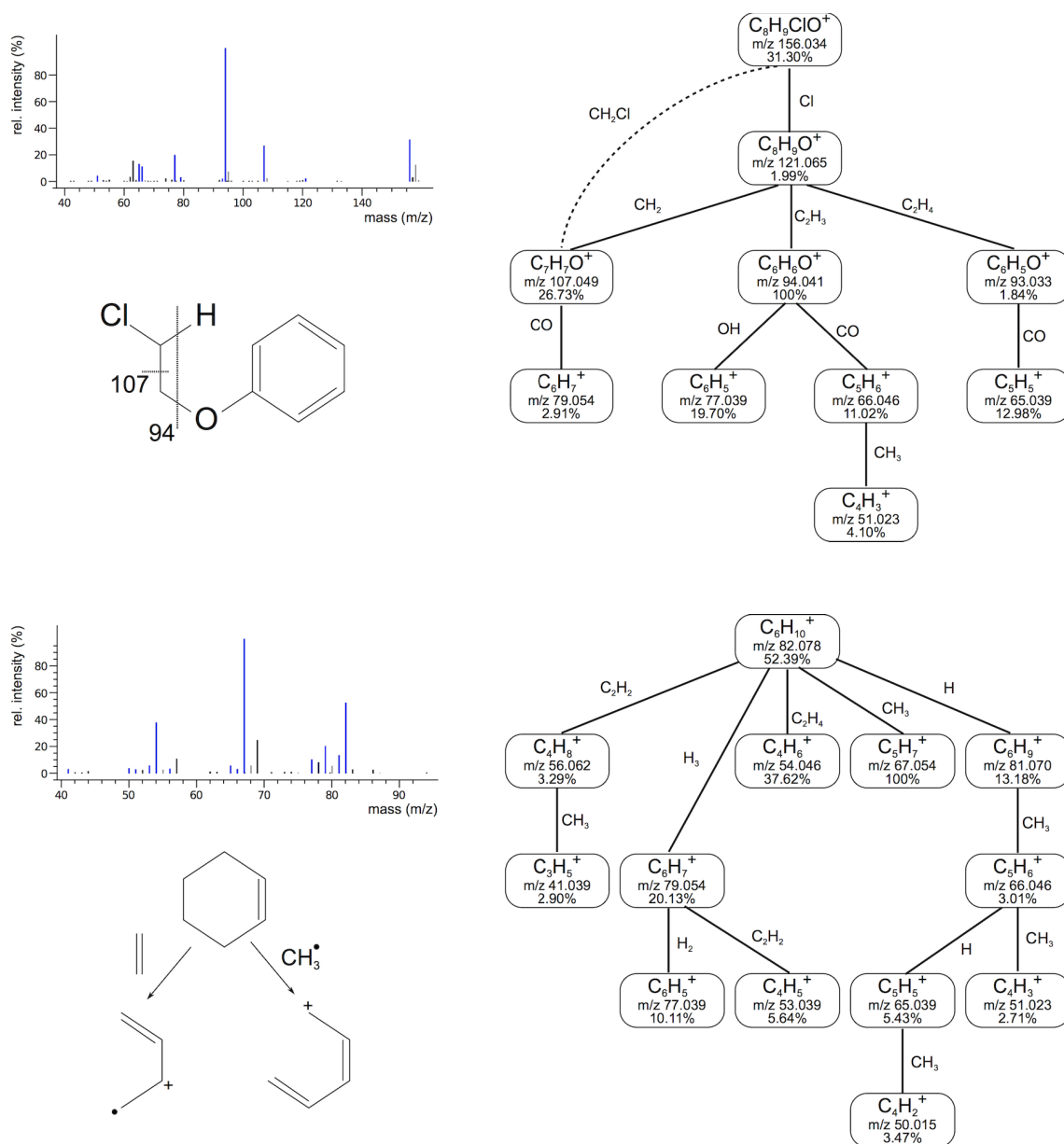


Figure 4.7: Molecular structures, measured mass spectra, and hypothetical fragmentation trees for (2-chloroethoxy) benzene (top) and cyclohexene (bottom). Fragment formulas, m/z values and intensities are given in the nodes. The trees serve as a basis for the further interpretation of the mass spectra. Peaks in the spectra are colored blue if an explanation was found. Peaks colored gray were filtered out as isotopes (details omitted). Note that radicals are not indicated in the trees.

Further reactions are in accordance with the formation of thermodynamically stable ions, such as in the loss of hydrogens to produce an aromatic moiety (82-79-77). The unidentified peak at m/z 69 corresponds to the major abundant fragment (CF_3^+) of heptacosane, used as the calibration substance. The peaks representing heptacosane fragments produced by EI can be removed from the spectrum by a background subtraction but some heptacosane ions occasionally remain. The 69 Da fragment was automatically disregarded by the method without specific instructions in the program to delete such common heptacosane fragments. This shows that the method can even deal with mass peaks that disrupt the spectrum of the test compound. This ability therefore enhances the analysis of mass spectra. As a result of comparing our fragmentation pattern analysis with textbook knowledge we conclude that the basic fragmentation principles reported in the literature are also modeled by our automated analysis. Small peaks that are not described by McLafferty like the formation of an aromatic system and further indicative fragmentations, can also be explained.

2-Ethylhexanal (CAS 123-05-7) The third example describes the fragmentation of a branched aldehyde (see Figure 4.8 (top)). The molecule 2-ethylhexanal undergoes a proton rearrangement followed by formation of a mass of 72 Da through two consecutive losses of C_2H_4 described by the contracted edges (128-100-72). This is a possible constitutive reaction proposed by McLafferty. The ion of m/z 72 fragments further to give m/z 57 by losing CH_3 . In conclusion the fragmentation tree confirms known mechanisms from the literature.

Para-nitro chlorobenzene (CAS 100-00-5) The fourth example is an aromatic nitrophenol [13] (see Figure 4.8 (bottom)). Oxygen or an NO radical can be lost, as can be seen in the edges (157-127) ($\text{M}-\text{NO}$)⁺ and (157-141) ($\text{M}-\text{O}$)⁺. In ortho nitro-aromatic moieties OH would be lost. Highly scored are the NO_2 loss occurring as a combination of two consecutive steps (157-141-111), and the HCl loss (111-75) forming aromatic breakdown products. Further reactions are chemically meaningful and mostly describe aromatic breakdown products and Cl or HCl losses. In this example of a nitro aromatic the loss of NO_2 induced by EI is represented by the combined edges of NO and O loss. We found the significant losses of the halogen atom (chlorine) and the fragmentation pattern, indicating aromatic breakdown products.

Ergosterol (CAS 57-87-4) This compound was selected as an example of a complex compound with high molecular mass. It is also a typical member of the steroid compound class (see Figure 4.9). It contains four condensed rings, an alcohol group located at the C-3 position and a side chain at C-17, which contains a double bond. The algorithm explores two pathways of fragmentation. The first of these was the abstraction of water (396-378) followed by α -cleavage of the side chain at C-17 represented by the edges (378-279-253). These edges show the presence of an alcohol and a labile side chain. The further cleavage reactions consist of cycloalkene and cycloalkane degradation by losses of CH_3 , C_2H_2 and C_2H_4 . These can possibly be explained by rDA reactions and α -cleavages common for unsaturated cyclic rings (see for example cyclohexene in Figure 4.7 (bottom)). A second pathway contains the loss of $\text{C}_3\text{H}_6\text{O}$ (396-338), known for cyclic alcohols [96]. This reaction is followed by the loss of C_6H_{14} . This loss possibly occurs through the previous opening of the B-ring to form a vitamin D2 structure as the intermediate, followed by the reaction of the unsaturated side chain double bond located on C-22. This reaction explains the stable

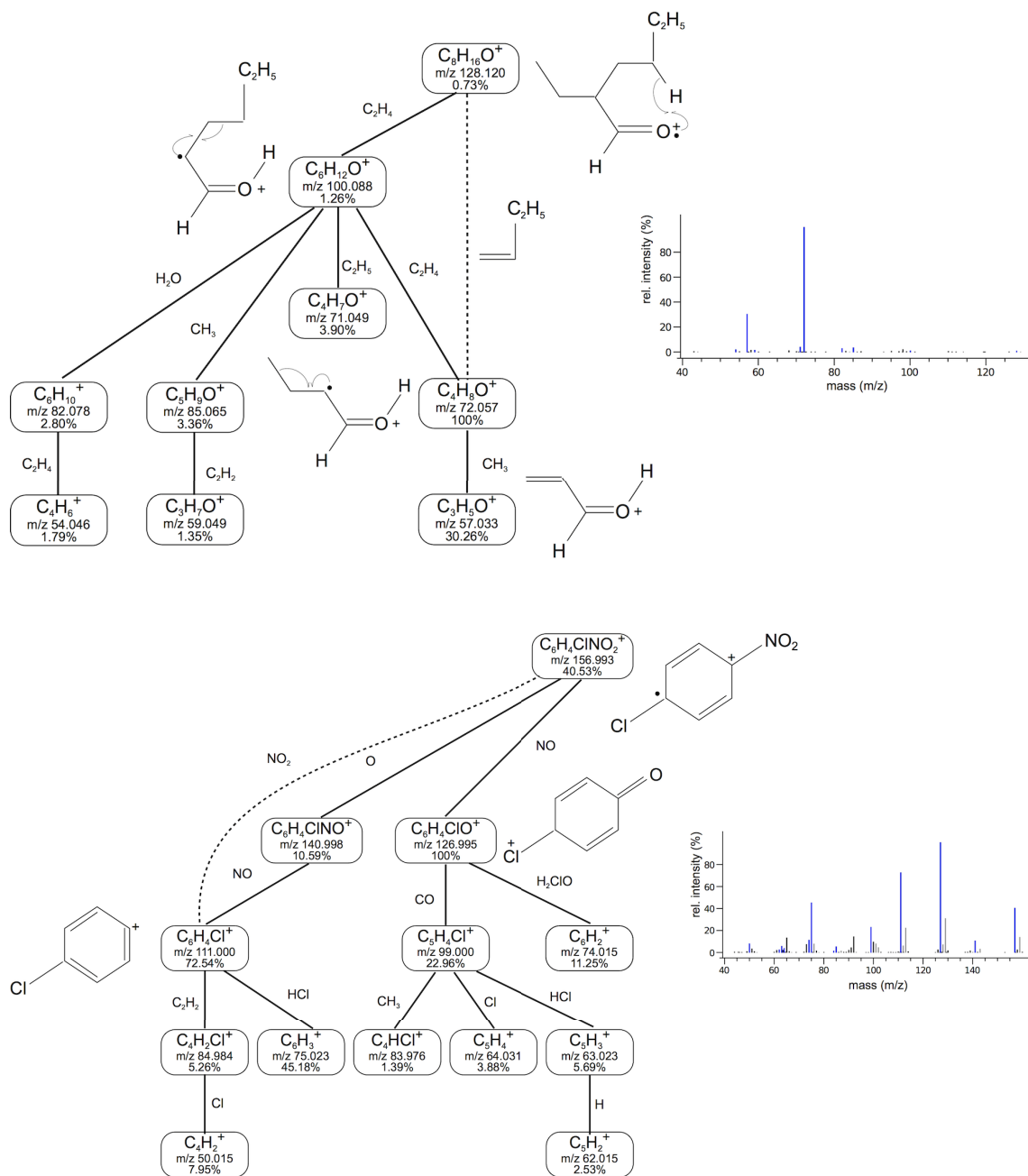


Figure 4.8: Molecular structures, measured mass spectra, and hypothetical fragmentation trees for 2-ethylhexanal (top) and para-nitro chlorobenzene (bottom). Fragment formulas, m/z values and intensities are given in the nodes. The trees serve as a basis for the further interpretation of the mass spectra. Peaks in the spectra are colored blue if an explanation was found. Peaks colored gray were filtered out as isotopes (details omitted). Note that radicals are not indicated in the trees.

C_6H_{14} loss (338-252). The possibility of the interconversion of ergosterol to Vitamin D2 is also supported by the finding of the H_2O loss in combination with a CH_3 loss represented in the edges of (396-378-363), previously shown by Zaretskii *et al.* [167]. All other reactions shown describe the degradation of the remaining polycyclic fragment. The peak at m/z 271 describes the ion resulting after cleavage of the side chain at C-17 without the loss of water ($\text{C}_{19}\text{H}_{27}\text{O}^+$). This peak was not identified since its mass deviation is twice that allowed for the respective intensity.

The hypothetical fragmentation trees of these example compounds and the remaining 45 compounds were then evaluated by the expert. The fragmentation trees annotated 1265 fragment peaks in total. From the 1006 peaks with relative intensity above 5 %, 865 (85 %) are explained in the trees. For the evaluation, the expert also used the molecular structure of the compounds, whereas our method for fragmentation tree computation uses *only* the MS data, plus the molecular formula of the compound. Our method identified all the important fragmentation reactions of the different compound classes. For example, the formation of stable onium ions is seen in all methyl ester compounds of fatty acids. The significant mass peaks of aromatic structural elements were also identified properly. Moreover, the method is capable of identifying losses that are very specific for a single compound or compound class and therefore not listed as a common loss, such as the $\text{C}_6\text{H}_6\text{N}$ loss of N-phenylbenzamide resulting in the formation of the stable aromatic onium ion.

It is important to note that the specific mechanisms of fragmentation are not reflected by the trees. For example, often only the combination of edges results in pathways that correspond to the true fragmentation. However, this is not a major set back since the relevant fragmentation can be constructed based on our analysis. For the central information on molecular ion and molecular formula, and the deduction of the compound class the plotted trees are sufficient.

4.2.5 Evaluation against MetFrag

For all underivatized compounds in the *measured dataset*, we compare the peak annotations from our hypothetical fragmentation trees to the *in silico* fragmentation using MetFrag¹ [166] (see Section 3.5.2). MetFrag has recently been extended to analyze EI fragmentation [130]. However, fragments resulting from non-trivial structural rearrangements are not covered by the bond disconnection approach.

For each compound, MetFrag is given the correct molecular formula and the complete peak list. We use MetFrag’s database search feature to search PubChem. For scoring the compounds, we use the positive [M] mode, an absolute error of 30 mDa and a relative error of 20 ppm. The number of possible hits and the ranks of the correct compounds are given in Table A.4 in the appendix. On average, the *in silico* fragmentation of all database hits (i.e. 328 per compound on average) takes 17.2 min per compound. For two steroids, we could achieve no identification, since the user session expired after two hours. Dicranin methyl ester could not be found in PubChem. We find that the rank of the correct identification in the output varies strongly, and appears to work worst for aldehydes.

We then compare the fragments predicted by MetFrag for the correct molecular structure with the peak annotations from our fragmentation trees (see Section 4.2.3). Again, we use a 1 % peak intensity cutoff and exclude peaks of isotopic distributions of the fragments,

¹<http://msbi.ipb-halle.de/MetFrag/>

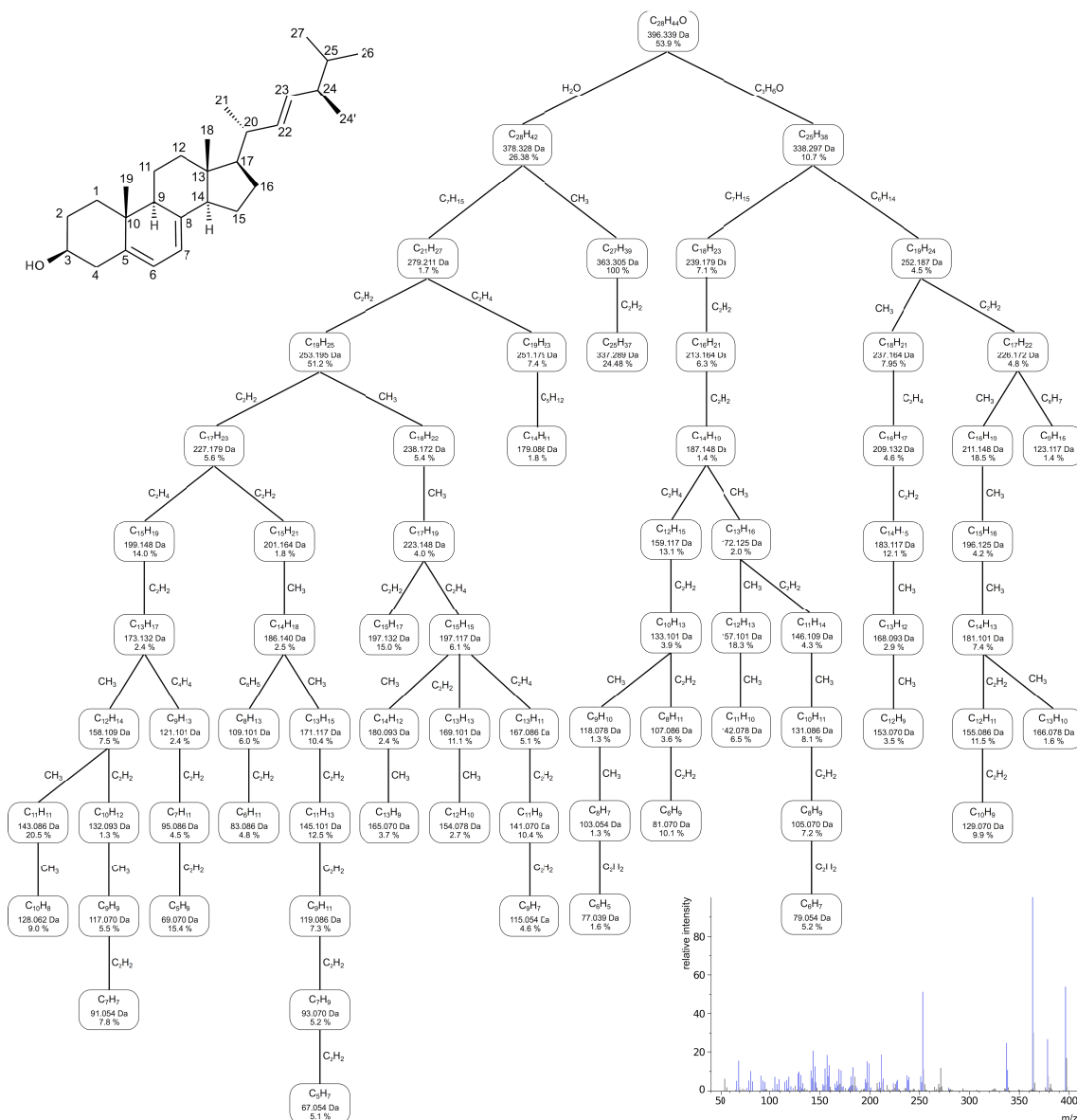


Figure 4.9: Hypothetical fragmentation tree of ergosterol. Fragment formulas, masses of the fragments (in Da) and intensities are given in the nodes. The tree serves as a basis for the further interpretation of the mass spectrum. Peaks in the spectrum are colored blue if an explanation was found. Peaks colored gray were filtered out as isotopes (details omitted). Note: radicals are not indicated in the tree.

resulting in 1488 peaks in total for the compounds that could be processed by both methods. Again, we use the positive [M] mode, an absolute error of 30 mDa and a relative error of 20 ppm. From the 1488 peaks, fragmentation trees explained 960 peaks and MetFrag explained 318 peaks in total (see Table A.4 in the appendix). For 240 peaks (75 % of the peaks annotated by MetFrag), both methods agreed on the molecular formula of a fragment.

Although a fragmenter is aware of the compound structure, structural rearrangements are still hard to predict. We stress that MetFrag is not designed to explain a maximum number of fragments. Enumerating more potential molecular fragments in MetFrag is possible and would explain more peaks, but has been shown to negatively impact the compound identification rate [166] (see Section 3.5.2).

5 Fragmentation Tree Alignment

MS analysis of similar compounds results in similar fragmentation trees. Rasche *et al.* [113] present local tree alignments for the automated comparison of fragmentation trees. They show that using the (annotated and more informative) fragmentation trees in applications such as database searching is superior to spectral comparison.

Rasche *et al.* [113] describe three workflows based on pairwise similarity scores between fragmentation trees: (1) clustering unknown compounds based solely on their fragmentation patterns; (2) predicting chemical similarity of molecules, since fragmentation pattern similarity is strongly correlated with chemical similarity; and (3) finding structurally similar compounds in a spectral library using *FT-BLAST* (Fragmentation Tree Basic Local Alignment Search Tool). *FT-BLAST* helps to overcome the limitations of spectral library search, as this tool, enables us to retrieve not only exact hits, but also similar compounds from a spectral database and to differentiate between true and spurious hits. Fragmentation tree alignments even allow for inter-dataset comparisons even for datasets measured on different instrument types [113]. All three applications have been discussed in detail by Rasche [111].

Performing these workflows on a large dataset requires tree alignments to be executed extremely fast. In this chapter, we present three exact algorithms for the alignment of fragmentation trees [60]. We modify the tree alignment algorithm from Jiang *et al.* [68] for edge similarities and local alignments, and show how to integrate JOIN nodes without increasing the worst-case running time. Further, we present a sparse dynamic programming algorithm and an Integer Linear Program (ILP) for the fragmentation tree alignment problem and evaluate all methods on real-world data.

In addition, we apply clustering based on fragmentation tree similarities to unknown metabolites from Icelandic poppy [113] to demonstrate the potential of the method in a real-world application.

5.1 Formal Problem Definition

Alignment of two labeled trees is a measure of similarity between those trees. For the automated comparison of fragmentation trees, we use pairwise *local alignments*. Local tree alignment is a generalization of local sequence alignment. A local tree alignment of two fragmentation trees contains those parts of the two trees where similar fragmentation cascades occurred (see Figure 5.1).

Tree alignments were introduced by Jiang *et al.* [68] who considered both *ordered* and *unordered* trees. They designed an algorithm for ordered trees, that is, the children of any node have a fixed order. This algorithm can be applied, for example, for RNA secondary structure comparison [86], as RNA structure trees are ordered. In contrast, the children of any node of a fragmentation tree are intrinsically unordered, as there is no sensible way to order the sub-fragments of some fragment. In this respect, fragmentation trees are more similar to phylogenetic trees than to RNA structure trees. Whereas efficient, polynomial-

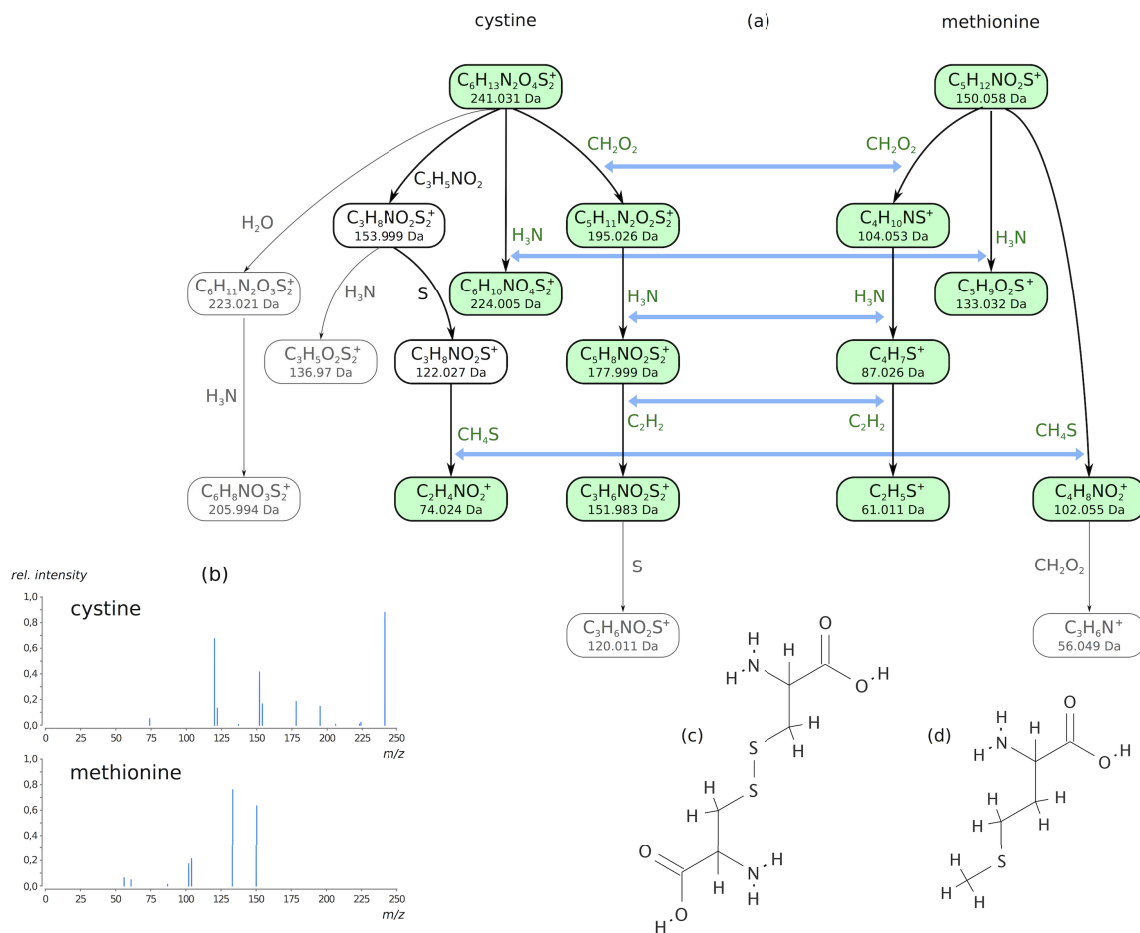


Figure 5.1: Optimal fragmentation tree alignment for cystine (11 losses) and methionine (6 losses) from the *Orbitrap* dataset (a). (b) Fragmentation mass spectra of cystine and methionine. The mass spectra do not share peaks. Molecular structures of cystine (c) and methionine (d). The molecular structures are not known to the alignment method. The alignment detects the common fragmentation path of formic acid-ammonia-ethylene losses and the separate ammonia branch. Additionally, it finds the methylthiol loss, which occurs at a later stage in cystine.

time algorithms exist for the alignment of ordered trees, the alignment of unordered trees is computationally hard, namely MAX SNP-hard [68]. This implies that there exists no Polynomial Time Approximation Scheme (PTAS) for the problem unless $P = NP$ [3]. In case both trees have fixed maximum out-degree, an optimum alignment can be computed via dynamic programming (DP) in polynomial time [68]. In comparison, computing the edit distance between two unordered trees remains MAX SNP-hard even for bounded degrees [169].

Fragmentation tree similarity is defined via edges (representing losses) and nodes (representing fragments). A tree alignment may contain matches, mismatches, insertions, and deletions, but respects the structure of the two trees. The similarity of two trees is then defined as the sum of the scores from all aligned edge pairs. Gap nodes and edges allow for insertions and deletions. We introduce JOIN nodes to account for missing nodes in one of the trees compared. Missing nodes result from missing peaks in one of the spectra.

Similarity of fragmentation trees can be measured by comparing losses (edges), comparing fragments (nodes) or even comparing both. For ease of presentation, we will concentrate on comparing losses only. However, scoring node pairs and scoring edge pairs are closely related. We can push an edge score into its end node, or we can pull a node score into its unique incoming edge. All algorithms presented here work both with node scoring and with edge scoring, as well as a combination thereof.

Let $T = (V, E)$ be a fragmentation tree with an edge labeling $\ell : E \rightarrow \mathcal{L}$. The following annotations are used throughout this chapter: Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be the two trees we want to align. For ease of reading, we sometimes call T_1 the *left tree* and T_2 the *right tree*. Let $p(v)$ be the parent node and $C(v)$ denote the set of children of any node v in T_1 or T_2 . We usually assume that u is a node of T_1 , and v a node of T_2 . For $i = 1, 2$, let $n_i := |V_i|$ be the number of nodes in T_i , and let d_i be the maximum out-degree in T_i . These maximum out-degrees will be of particular interest to us, as the running time of our dynamic programming grows exponentially in d_1, d_2 . Let $\delta = \min\{d_1, d_2\}$ and $\Delta = \max\{d_1, d_2\}$.

We define a similarity function $\sigma : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}$ for pairs of losses, which implies a similarity function $\sigma : E_1 \times E_2 \rightarrow \mathbb{R}$ between edges of the two trees T_1, T_2 via $\sigma(e_1, e_2) = \sigma(\ell(e_1), \ell(e_2))$. Furthermore, we introduce a JOIN operator (see Figure 5.2(b)): Given a path p_1 in T_1 of length two, let e_1, e'_1 be the edges of p_1 . We can assign a loss to p_1 by adding the corresponding losses $\ell(e_1) + \ell(e'_1) \in \mathcal{L}$. This means taking the sum of the respective compomers or the additive union of the corresponding multisets. We then assign a similarity between p_1 and any edge e_2 of T_2 as $\sigma(p_1, e_2) = \sigma(\ell(e_1) + \ell(e'_1), \ell(e_2))$. Analogously, we can define a similarity for paths of length two in T_2 . Obviously, this can be generalized to paths of arbitrary lengths but here, we will limit ourselves to paths of length two. For joining nodes in the alignment, we assume *homogeneous join costs*: the penalty for joining a node is $\sigma_{\text{join}} \leq 0$, independent of the node or edge that we want to join. Formally, this allows us to focus on the important aspects of our algorithms, and omit some technical details. Practically, we currently see no biologically reasonable way to assign different scores to different join nodes, as these usually correspond to the non-detection of a peak in one of the mass spectra.

As mentioned above, a node scoring can be easily transformed to an edge scoring by pulling all node scores into their unique incoming edges. However, the root node has no incoming edge. Thus, for edge scoring the root node is not considered. Nevertheless, the two scorings can be easily combined by introducing a particular *root scoring* $\sigma^* : V_1 \times V_2 \rightarrow \mathbb{R}$ for the root nodes of the alignment.

Let T_1, T_2 be two trees. We define a *global alignment* \mathcal{A} of T_1, T_2 as follows [68]: \mathcal{A} is a tree where nodes are labeled with pairs from $(V_1 \cup \{-\}) \times (V_2 \cup \{-\})$. Here, ‘-’ is the *gap symbol* (see Figure 5.2(a)). If we restrict labels of \mathcal{A} to the first coordinate and contract all edges that end in a node labeled ‘-’, we end up with the tree T_1 ; if we do the same for the second coordinate we end up with the tree T_2 . (In fact, we have to replace the nodes of the restricted trees by their labels, we omit the simple technical details.) We say that \mathcal{A} is a *local alignment* if the trees originating from contracting gap-edges are induced subtrees of T_1 and T_2 , respectively.

Different from Jiang *et al.* [68], we want to score an alignment based on the *edges* of the two trees. To this end, for any node a of \mathcal{A} but the root, let $e_1(a)$ be the unique edge in T_1 that ends in the first coordinate of the label of a , and let $e_2(a)$ be the unique edge in

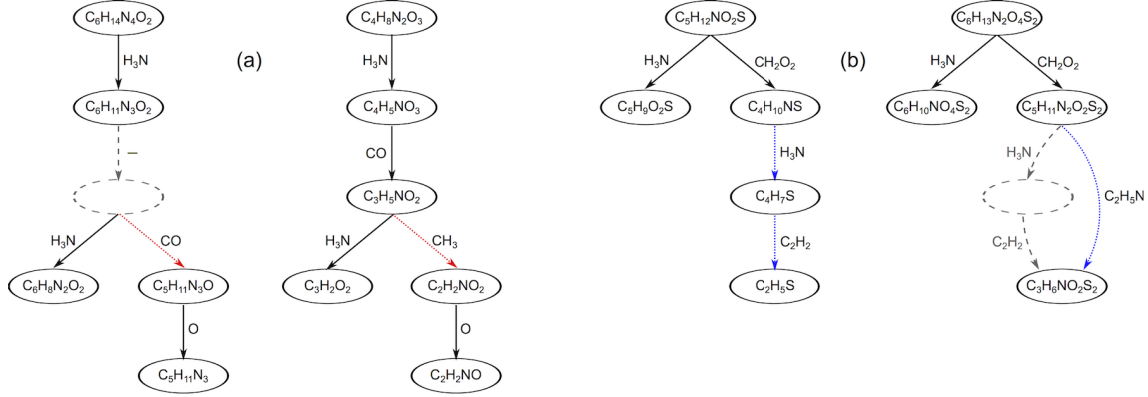


Figure 5.2: Two alignments of fragmentation trees based on edge similarities. (a) A gap (–) is introduced for the missing CO loss in the left tree (gray dashed edge and node). Losses CO and CH_3 are aligned by a mismatch (red dotted edges). (b) In the right tree the fragment after losing H_3N is missing (gray dashed edges and node), while the fragment after further loss of C_2H_2 is observed. To account for missing fragments we introduce the JOIN operation. It allows to align the two successive losses H_3N and C_2H_2 in the left tree to a single loss C_2H_5N in the right tree (blue dotted edges). Fragments may be missing because the corresponding peak was not detected, for example.

T_2 that ends in the second coordinate of the label of a . In case no such edge exists, we assume $e_1(a) = '-'$ or $e_2(a) = '-'$, respectively. Now, we define the *score* of \mathcal{A} as

$$\sum_{\text{non-root node } a \text{ of } \mathcal{A}} \sigma(e_1(a), e_2(a)).$$

We define $\sigma(T_1, T_2)$ as the maximum score of a local alignment of T_1 and T_2 .

5.2 Alignment Algorithms

Performing the workflows proposed by Rasche *et al.* [113] requires all-against-all fragmentation tree alignments to obtain a matrix of pairwise fragmentation tree similarities (see Section 5.4.4). On large datasets this means that tree alignments have to be performed very often (see Table 5.1) and therefore executed extremely fast. In this section, we present three exact algorithms for the alignment of fragmentation trees [60].

5.2.1 Dynamic Programming

We now present a dynamic programming algorithm for the computation of optimum fragmentation tree alignments that has reasonable running time in practice. Our algorithm is a modification of an algorithm by Jiang *et al.* [68] for computing global alignments of unordered trees. An informal version of the algorithm was presented in [113], but no correctness proof or running time analysis for the algorithm was given. Note that in this older version, only one child node is allowed to be joined with its parent, all other children are discarded. The results presented in Section 5.4 are based on this older version.

Fragmentation trees usually have comparatively small out-degree: fragments rarely have more than, e.g., five child fragments. We can limit the inevitable exponential part of the running time to this out-degree.

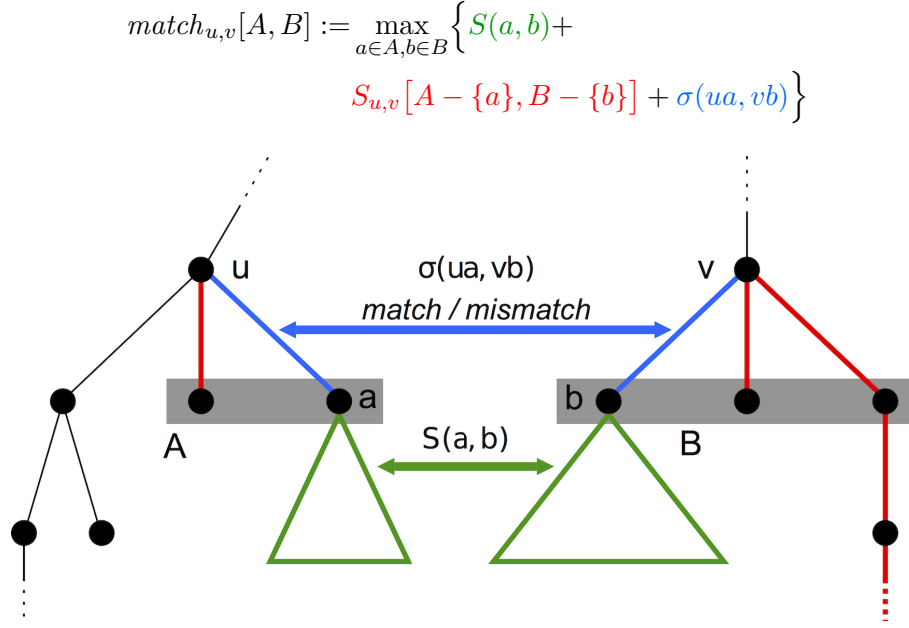


Figure 5.3: Representation of the *match* recurrence of the dynamic programming algorithm (see Recurrence 5.3). $match_{u,v}[A, B]$ is the best score to match the two nodes u, v , such that maximally the children A of u and B of v are used. To do so, we have to match at least one outgoing edge from u with one outgoing edge from v . The score of matching edge ua to edge vb , is the sum of the score of the local alignment of the two subtrees rooted in a and b (green), the score of the optimum local alignment of the remaining children (red) and the (mis-)match score of the two edges (blue).

We use dynamic programming (DP) to compute the maximum score $\sigma(T_1, T_2)$ of a local alignment between two trees T_1, T_2 . Let $S(u, v)$ be the maximum score of a local alignment of two subtrees of T_1, T_2 , where the subtree of T_1 is rooted in u , and the subtree of T_2 is rooted in v . For $A \subseteq C(u)$ and $B \subseteq C(v)$, we define $S_{u,v}[A, B]$ to be the score of an optimum local alignment of subtrees rooted in u and v , respectively, such that *maximally* the children A of u and B of v are used in the alignment. Clearly, $S(u, v) = S_{u,v}[C(u), C(v)]$. Furthermore, we have $S_{u,v}[A, \emptyset] = S_{u,v}[\emptyset, B] = 0$ for all A, B . When all $S(u, v)$ are known, we can compute the maximum score of a local alignment of T_1, T_2 as

$$\sigma(T_1, T_2) = \max_{u \in T_1, v \in T_2} S(u, v). \quad (5.1)$$

We present a recurrence for the computation of $S_{u,v}[A, B]$ (see Figures 5.3 and 5.4). We initialize $S_{u,v}[A, B] = 0$ for $A = \emptyset$ or $B = \emptyset$. Recall that T_1 is the left tree and T_2 is the right tree. In the recurrence, we distinguish three cases, namely *match* (including mismatches), *deletion left*, or *deletion right*, where the latter two are symmetric. For non-empty sets $A \subseteq C(u)$ and $B \subseteq C(v)$, we set

$$S_{u,v}[A, B] = \max \left\{ 0, match_{u,v}[A, B], deleteL_{u,v}[A, B], deleteR_{u,v}[A, B] \right\} \quad (5.2)$$

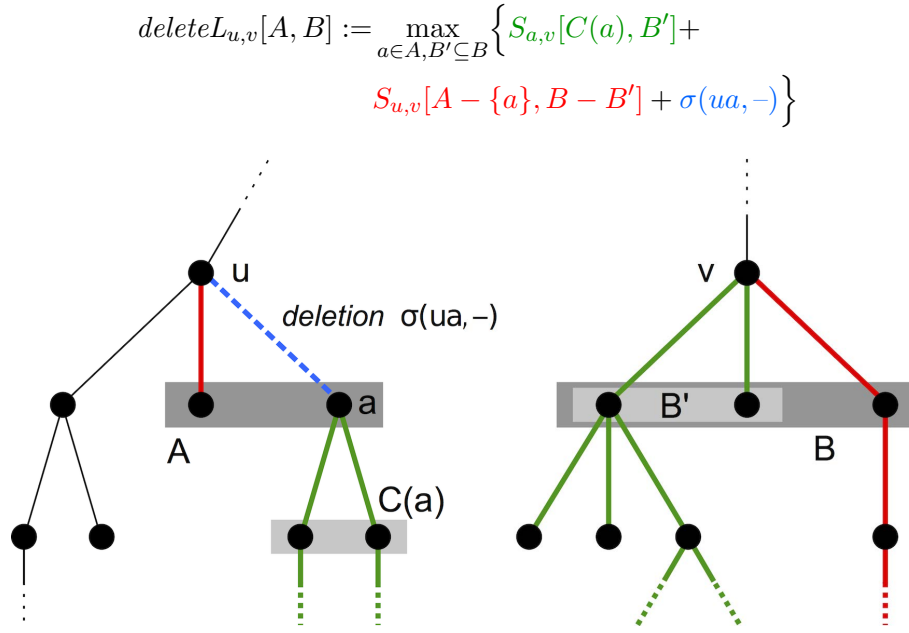


Figure 5.4: Representation of the delete_L recurrence of the dynamic programming algorithm (see Recurrence 5.3). $\text{delete}_{u,v}[A, B]$ is the best score for deleting edge ua , such that maximally the children A of u and B of v are used. A subset $B' \subseteq B$ of the children of v can now be matched to the children of a . The resulting score is the sum of aligning the children of a to some children of v that are in B' (green), aligning the siblings of a that are in $A - \{a\}$ to the remaining children of v that are in $B - B'$ (red) and the costs for deleting edge ua , that is, $\sigma(ua, -)$ (blue).

where we define

$$\begin{aligned} \text{match}_{u,v}[A, B] &:= \max_{a \in A, b \in B} \left\{ S(a, b) + S_{u,v}[A - \{a\}, B - \{b\}] + \sigma(ua, vb) \right\} \\ \text{delete}_{u,v}[A, B] &:= \max_{a \in A, B' \subseteq B} \left\{ S_{a,v}[C(a), B'] + S_{u,v}[A - \{a\}, B - B'] + \sigma(ua, -) \right\} \\ \text{delete}_{R_{u,v}}[A, B] &:= \max_{A' \subseteq A, b \in B} \left\{ S_{u,b}[A', C(b)] + S_{u,v}[A - A', B - \{b\}] + \sigma(-, vb) \right\} \end{aligned} \tag{5.3}$$

Here, $\sigma(ua, vb)$ denotes the score of the losses attached to edges ua and vb , and $\sigma(ua, -), \sigma(-, vb)$ accordingly. Recurrence (5.3) is the obvious modification of the recurrence presented in Jiang *et al.* [68] which was designed for global alignments and node similarities.

Merging two losses in T_1 or T_2 requires two additional symmetric cases, namely *join left* and *join right* for merging in tree T_1 or T_2 , respectively. To speed up computations, we add

an additional PREJOIN case for nodes that will be joined in the alignment (see Figure 5.5). We set

$$S_{u,v}[A, B] = \max \left\{ 0, \text{match}_{u,v}[A, B], \right. \\ \left. \text{delete}L_{u,v}[A, B], \text{delete}R_{u,v}[A, B], \right. \\ \left. \text{join}L_{u,v}[A, B], \text{join}R_{u,v}[A, B] \right\} \quad (5.4)$$

where we define, in addition to Recurrence (5.3),

$$\begin{aligned} \text{prejoin}L_{u,v}[A, B] &:= \max_{a \in A, b \in B} \left\{ S(a, b) + \right. \\ &\quad \text{prejoin}L_{u,v}[A - \{a\}, B - \{b\}] + \\ &\quad \left. \sigma(p(u)a, vb) + \sigma_{\text{join}} \right\} \\ \text{join}L_{u,v}[A, B] &:= \max_{a \in A, B' \subseteq B} \left\{ \text{prejoin}L_{u,v}[C(a), B'] + \right. \\ &\quad \left. S_{u,v}[A - \{a\}, B - B'] \right\} \end{aligned} \quad (5.5)$$

Here, $\sigma(p(u)a, vb)$ is the score for the combined losses on the path from $p(u)$ to a with the loss of edge vb . Recall that $\sigma_{\text{join}} \leq 0$ is the penalty for joining a node. Again, we initialize $\text{join}L_{u,v}[A, \emptyset] = \text{join}L_{u,v}[\emptyset, B] = 0$ for all A, B . Analogously to Recurrence (5.5), we can define recurrences for $\text{prejoin}R_{u,v}[A, B]$ and $\text{join}R_{u,v}[A, B]$.

For *bottom-up* DP [140], we have to find an order in which the entries of the DP tables can be filled. Computation of $\text{match}_{u,v}[A, B]$, $\text{delete}L_{u,v}[A, B]$, and $\text{delete}R_{u,v}[A, B]$ only accesses entries $S_{u',v'}[A', B']$ such that $u' \in \{u\} \cup C(u)$ and $v' \in \{v\} \cup C(v)$. By processing nodes in postorder, we ensure that all $S_{u',v'}[A', B']$ are previously computed for $(u', v') \neq (u, v)$. For the remaining case, we iterate $|A| + |B| = 0, 1, \dots, |C(u)| + |C(v)|$. Similar arguments hold for the computation of JOIN and PREJOIN nodes.

Theorem 1. *Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two trees, $\sigma : E_1 \cup \{-\} \times E_2 \cup \{-\} \rightarrow \mathbb{R}$ a scoring function between edge pairs, and $\sigma_{\text{join}} \in \mathbb{R}$ the penalty for joining a node. For $i = 1, 2$, set $n_i := |V_i|$, and let d_i be the maximum out-degree in T_i . The maximum score $\sigma(T_1, T_2)$ of a local alignment of T_1, T_2 can be computed in $O(3^\Delta \cdot 2^\delta \cdot \delta n_1 n_2)$ using Recurrence (5.4) and Equation (5.1), where $\Delta := \max\{d_1, d_2\}$ and $\delta := \min\{d_1, d_2\}$.*

Proof. The running time for computing $S_{u,v}[A, B]$ is dominated by the computation of $\text{delete}L_{u,v}[A, B]$ and $\text{delete}R_{u,v}[A, B]$, as well as $\text{join}L_{u,v}[A, B]$ and $\text{join}R_{u,v}[A, B]$. To compute $\text{delete}R_{u,v}[A, B]$ for all $A \subseteq C(u)$ and $B \subseteq C(v)$ we have to iterate over all $A' \subseteq A$ and all $b \in B$. Iterating over all subsets $A' \subseteq A \subseteq C(u)$ needs 3^{d_u} time, where $d_u = |C(u)|$. Iterating over all $b \in B \subseteq C(v)$ needs $2^{d_v} \cdot d_v$ time, where $d_v = |C(v)|$. This leads to an overall running time of $O(3^{d_u} \cdot 2^{d_v} \cdot d_v)$ for the computation of $\text{delete}R_{u,v}[A, B]$ and $O(2^{d_u} \cdot 3^{d_v} \cdot d_u)$ for $\text{delete}L_{u,v}[A, B]$. For $\text{join}L_{u,v}[A, B]$ and $\text{join}R_{u,v}[A, B]$, the proof is similar. \square

The proof of the theorem is based on the following lemma:

Lemma 1. *Computing $S_{u,v}[A, B]$ for all $A \subseteq C(u)$ and $B \subseteq C(v)$ is possible using Recurrence (5.4) in $O(3^{d_u} \cdot 2^{d_v} \cdot d_v + 2^{d_u} \cdot 3^{d_v} \cdot d_u)$ time, where $d_u = |C(u)|$ and $d_v = |C(v)|$.*

$$\begin{aligned}
\text{join}L_{u,v}[A, B] &:= \max_{a \in A, B' \subseteq B} \left\{ \text{prejoin}L_{a,v}[C(a), B'] + \right. \\
&\quad \left. S_{u,v}[A - \{a\}, B - B'] \right\} \\
\text{prejoin}L_{a,v}[C(a), B'] &:= \max_{x \in C(a), b \in B'} \left\{ S(x, b) + \sigma_{\text{join}} + \right. \\
&\quad \left. \text{prejoin}L_{a,v}[C(a) - \{x\}, B' - \{b\}] + \sigma(p(a)x, vb) \right\}
\end{aligned}$$

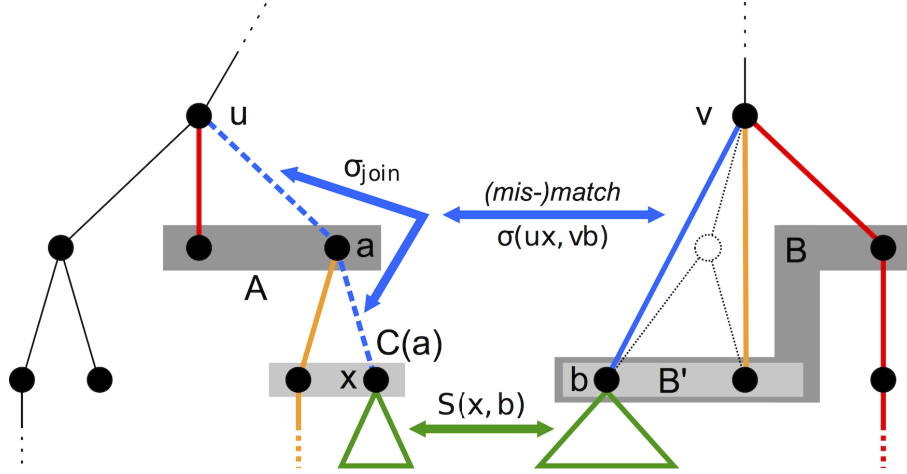


Figure 5.5: Representation of the *joinL* recurrence of the dynamic programming algorithm (see Recurrence 5.5). To speed up computations, we add an additional PREJOIN case which considers how to match a subset $B' \subseteq B$ of the children of v to the children of a , taking into account: the score for merging edges ua and ax in the left tree and matching the joined edge to edge vb in the right tree (blue); the score of the aligned subtrees rooted in x and b (green); and the PREJOIN score for matching the remaining children of a , that is, $C(a) - \{x\}$ to the remaining nodes from $B' - \{b\}$ (orange). The *joinL* case further includes the score of aligning the siblings of a that are in $A - \{a\}$ to the remaining children of v that are in $B - B'$ (red).

Proof. To prove the correctness of Recurrence 5.4 we have to distinguish five cases: nodes u, v are matched, u is deleted, v is deleted, u is merged with at least one of its children or v is merged with at least one of its children. We know $S_{u,v}[A, \emptyset] = S_{u,v}[\emptyset, B] = 0$.

Matching u, v is covered by $\text{match}_{u,v}[A, B]$ (see Figure 5.3). To match two nodes u, v we have to match at least one outgoing edge from u with one outgoing edge from v and therefore one child $a \in A$ with one child $b \in B$. For each $a \in A$ and $b \in B$, $S(a, b)$ is the maximum score of the local alignment of the two subtrees rooted in a and b , $\sigma(ua, vb)$ is the score of matching ua to vb . Furthermore $S_{u,v}[A - \{a\}, B - \{b\}]$ is the score of the optimum local alignment of the remaining children. The sum is the optimum score for $S_{u,v}[A, B]$ when matching ua with vb . Since at least one outgoing edge from u has to be matched to one outgoing edge from v , this is in fact the maximum over all children $a \in A$ and $b \in B$.

The deletion of an outgoing edge from u is covered by $\text{delete}L_{u,v}[A, B]$ (see Figure 5.4). When deleting edge ua from the left tree, we have to bipartition the children of v depending on whether they match with children of u or of a . Therefore we have to iterate over all subsets $B' \subseteq B$. The maximum score of a local alignment with subtree rooted in a and v ,

using all children of a and the children in $B' \subseteq B$ from v , is already given by $S_{a,v}[C(a), B']$. The remaining children $B - B'$ from v are matched with the children from u , where the maximum score is also already given by $S_{u,v}[A - \{a\}, B - B']$. Furthermore, we have to add the costs for deleting an edge given by $\sigma(ua, -)$. The sum of these three values is maximized over all bipartitions of B . Deleting a node u means deleting one of its outgoing edges, so we further have to maximize over all children $a \in A$.

The reasoning when deleting v is analogous.

Merging u with one of its successors is covered by $joinL_{u,v}[A, B]$ (see Figure 5.5). When merging node u with node a in the left tree, we have to merge edge ua at least with one outgoing edge from a and match this merged edge to an outgoing edge from v in the right tree. Analogous to *deleteL* We have to bipartition the children of v depending on whether they match with children of u or of a . The maximum score of a local alignment rooted in u and v , where the subset of children $B - B'$ from v is matched with the children $A - \{a\}$ from u is already given by $S_{u,v}[A - \{a\}, B - B']$. The remaining children B' from v are matched with children $C(a)$. This score is already stored in the *PREJOIN* case $prejoinL_{a,v}[C(a), B']$. The sum of these two values is maximized over all bipartitions of B . Merging a node u means merging one of its outgoing edges with a subsequent edge, so we further have to maximize over all children $a \in A$.

For the $prejoinL_{a,v}[C(a), B']$ case, we have to merge at least one outgoing edge from a with edge ua and match it to one outgoing edge from v . To do so, we have to consider the score of this match $\sigma(p(a)x, vb)$ (that is $\sigma(ux, vb)$), the join cost σ_{join} , and the the score $S(x, b)$ of the aligned subtrees rooted in x and b . Furthermore the siblings of x (that is $C(a) - \{x\}$) have to be matched to siblings of b from the subset B' , which is already stored in $prejoinL_{a,v}[C(a) - \{x\}, B' - \{b\}]$. Since edge $p(a)a$ has to be merged with at least one outgoing edge from a and matched to one outgoing edge from v , we have to maximize over all children $x \in C(a)$ and $b \in B'$.

The reasoning when joining v with at least one of its children is analogous.

Computing the maximum over all five cases and 0 results in the maximum score for $S_{u,v}[A, B]$, that is, the score of an optimum local alignment with subtree rooted in u and v , respectively, such that at most the children A of u and B of v are used in the alignment. \square

Similarly to Theorem 1, we can show that any pairwise tree alignment that does not take joining nodes into account, can also be computed in this time. We leave out the straightforward details.

Theorem 2. *A pairwise unordered tree alignment (global or local, scoring nodes or edges or both, with similarities or costs) of rooted trees T_1, T_2 can be computed in $O(3^\Delta \cdot 2^\delta \cdot \delta n_1 n_2)$ time. Here, n_i is the number of nodes in tree T_i , and d_i is the maximum out-degree in T_i , for $i = 1, 2$; furthermore, $\Delta := \max\{d_1, d_2\}$ and $\delta := \min\{d_1, d_2\}$.*

We conjecture that running time of the DP can be improved to $O(2^{d_1+d_2} \cdot \text{poly}(d_1, d_2) n_1 n_2)$ using the Möbius transform [5], but this appears to be of theoretical interest only.

5.2.2 Sparse Dynamic Programming

Applying the above algorithm to real-world instances of aligning fragmentation trees, one can see that $S(u, v) = 0$ holds for many node pairs u, v . This can be attributed to two

factors: First, we are computing local alignments, so we can always choose to end the subtrees that are part of the alignment in the nodes u, v . Second, there are many different labels found at the edges (or nodes) of a fragmentation tree. A reasonable scoring scheme will assign negative scores to most non-matching edge (or node) labels, so it is rather the exception than the rule that we can find two nodes u, v with $S(u, v) > 0$.

The idea is to “sparsify” our DP tables by storing only those table entries with positive values. Thereby, we face the following fact: If $S_{u,v}[A, B] > 0$ for $A \subseteq C(u)$ and $B \subseteq C(v)$ then $S_{u,v}[A', B'] > 0$ holds for all supersets A', B' with $A \subseteq A' \subseteq C(u)$ and $B \subseteq B' \subseteq C(v)$. Thus, as soon as we have one non-zero entry in the table an exponentially large part of the table will be filled with non-zero entries, too.

To negate this rather unfortunate effect, we modify our DP as follows: For $A \subseteq C(u)$ and $B \subseteq C(v)$, we define $S'_{u,v}[A, B]$ to be the score of an optimum local alignment with subtrees rooted in u and v , respectively, such that *exactly* the children A of u and B of v are used in the local alignment. If no such alignment exists, we set $S'_{u,v}[A, B] = -\infty$. Then $S'_{u,v}[\emptyset, \emptyset] = 0$, but for all $A, B \neq \emptyset$ we have $S'_{u,v}[A, \emptyset] < 0$, $S'_{u,v}[\emptyset, B] < 0$. Clearly,

$$S(u, v) = \max_{A \subseteq C(u), B \subseteq C(v)} S'_{u,v}[A, B]. \quad (5.6)$$

We need one more trick in our recurrence: in Recurrence (5.3) we have accessed entries $S_{a,v}[C(a), B']$ and $S_{u,b}[A', C(b)]$, but this is not possible for the table S' as the optimum alignments might not use all the children of a or b . To this end, we introduce

$$\begin{aligned} S'_{u,v}[A, *] &:= \max_{B' \subseteq C(v)} \{S'_{u,v}[A, B']\}, \\ S'_{u,v}[* , B] &:= \max_{A' \subseteq C(u)} \{S'_{u,v}[A', B]\}, \end{aligned}$$

for the maximum over all subsets of $C(v)$ or $C(u)$, respectively. For non-empty sets $A \subseteq C(u)$ and $B \subseteq C(v)$, we set

$$\begin{aligned} S'_{u,v}[A, B] = \max \{ & match'_{u,v}[A, B], \\ & deleteL'_{u,v}[A, B], deleteR'_{u,v}[A, B], \\ & joinL'_{u,v}[A, B], joinR'_{u,v}[A, B] \} \end{aligned} \quad (5.7)$$

which, compared to Recurrence (5.4), misses the lower bound 0 and uses the definitions:

$$\begin{aligned} match'_{u,v}[A, B] &:= \max_{a \in A, b \in B} \{ S(a, b) + \\ & \quad S'_{u,v}[A - \{a\}, B - \{b\}] + \sigma(ua, vb) \} \\ deleteL'_{u,v}[A, B] &:= \max_{a \in A, B' \subseteq B} \{ S'_{a,v}[* , B'] + \\ & \quad S'_{u,v}[A - \{a\}, B - B'] + \sigma(ua, -) \} \\ deleteR'_{u,v}[A, B] &:= \max_{A' \subseteq A, b \in B} \{ S'_{u,b}[A', *] + \\ & \quad S'_{u,v}[A - A', B - \{b\}] + \sigma(-, vb) \} \end{aligned} \quad (5.8)$$

For the further JOIN recurrences, we only concentrate on the left tree. The definition of $prejoinL'_{u,v}[A, *]$ and the JOIN recurrences at the right tree are analogous.

$$\begin{aligned}
prejoinL'_{u,v}[A, B] &:= \max_{a \in A, b \in B} \left\{ S(a, b) + \right. \\
&\quad \left. prejoinL'_{u,v}[A - \{a\}, B - \{b\}] + \right. \\
&\quad \left. \sigma(p(u)a, vb) + \sigma_{\text{join}} \right\} \\
prejoinL'_{u,v}[* , B] &:= \max_{A' \subseteq A} \left\{ prejoinL'_{u,v}[A', B] \right\} \\
joinL'_{u,v}[A, B] &:= \max_{a \in A, B' \subseteq B} \left\{ prejoinL'_{u,v}[* , B'] + \right. \\
&\quad \left. S'_{u,v}[A - \{a\}, B - B'] \right\}
\end{aligned} \tag{5.9}$$

To summarize, the central point is that we do not have to store any entries with $S'_{u,v}[A, B] \leq 0$: Such entries will never lead to an optimum alignment, as we are better off removing all nodes A, B , plus everything below these nodes from the alignment. The only exception to this rule is that we store the entry $S'_{u,v}[\emptyset, \emptyset] = 0$. Furthermore, we do not have to store entries $S'_{u,v}[A, B]$ if there exist subsets $A' \subseteq A, B' \subseteq B$ with $(A', B') \neq (A, B)$ such that $S'_{u,v}[A, B] \leq S'_{u,v}[A', B']$. In this case, we can replace an alignment that uses children A, B of u, v , by an alignment that uses only children A', B' and has better or equal score. We say that an entry $S'_{u,v}[A, B]$ is *dominated* by entry $S'_{u,v}[A', B']$. For a scoring scheme that assigns negative scores for non-matching edge (or node) labels, large parts of the tables have negative scores or are dominated by another entry. We do not actually have to *forbid* that dominated entries are stored, as they do not interfere with our computations; rather, we are free to leave out dominated entries when we encounter them.

The resulting tables $S'_{u,v}$ are sparsely populated, and for many nodes u, v there are no entries with $S'_{u,v}[A, B] > 0$. We can reduce the memory consumption of the method using hash maps instead of arrays. Hash map implementations like Cuckoo hashing [105] or Hopscotch hashing [48] can carry out all operations in constant (amortized) time. In practice, we find that memory consumption is usually not prohibitive. In this case, we can use lazy arrays which are not allocated until a first entry is stored.

Resolving the recurrences. Now, it is time for our final trick: Instead of computing the scores using Recurrences (5.7–5.9), we apply a *successive approximation procedure* similar to Dijkstra’s Algorithm for shortest paths [140]. That is, instead of “pulling” scores from previously calculated entries, we “push” scores from entries that have been finalized. For example, assume that we have finalized the computation of some entry $S'_{u,v}[A, B]$ for fixed $A \subseteq C(u)$ and $B \subseteq C(v)$. Also assume that $S'_{u,v}[A, B] > 0$ as otherwise, $S'_{u,v}[A, B]$ is dominated by $S'_{u,v}[\emptyset, \emptyset] = 0$. Then, Recurrence (5.8) tells us that we can update other entries of the table accordingly: If $S'_{u,v}[A, B] > S'_{u,v}[* , B]$ (which we assume to be incompletely calculated so far) then $S'_{u,v}[* , B] \leftarrow S'_{u,v}[A, B]$. Similarly, if $S'_{u,v}[A, B] > S'_{u,v}[A, *]$ then $S'_{u,v}[A, *] \leftarrow S'_{u,v}[A, B]$, and if $S'_{u,v}[A, B] > S(u, v)$ then $S(u, v) \leftarrow S'_{u,v}[A, B]$. Regarding the recurrence for $match'$, we iterate over all $a \in C(u) \setminus A$ and $b \in C(v) \setminus B$: If $match'_{u,v}[A \cup \{a\}, B \cup \{b\}] < S(a, b) + S'_{u,v}[A, B] + \sigma(ua, vb)$ then update it accordingly. If $match'_{u,v}[A \cup \{a\}, B \cup \{b\}] \leq match'_{u,v}[A, B]$ then the entry $match'_{u,v}[A \cup \{a\}, B \cup \{b\}]$ is dominated and we can remove it from the hash map.

For all other cases, similar updates can be performed, which we only sketch here: For $deleteL'$, we iterate over all $a \in C(u) \setminus A$ and $B' \subseteq C(v) \setminus B$; if $deleteL'_{u,v}[A \cup \{a\}, B \cup B'] < S'_{a,v}[* , B'] + S'_{u,v}[A, B] + \sigma(ua, -)$ then update it accordingly. Updates have to be performed as soon as an entry is finalized, that is, it cannot be changed by any future modifications. Finding finalized entries is similar to the order of computations in the previous section; we omit the technical details.

The above algorithm has exactly the same worst-case running time complexity as the initial recurrence from Section 5.2.1. But in practice, we can get even faster, at least in cases where the arrays are very sparse: To this end, finalizing some entry $deleteL'_{u,v}[A, B]$ triggers updates for all subsets $B' \subseteq C(v) \setminus B$. But only those B' can lead to relevant updates where $S'_{a,v}[* , B'] > 0$ holds. Otherwise, the updated entry will be dominated by $S'_{a,v}[* , \emptyset] = 0$. If we iterate over the hash map for those B' with $S'_{a,v}[* , B'] > 0$ then the worst-case running time increases to $O(4^\Delta \cdot 2^\delta \cdot \delta n_1 n_2)$, assuming constant time access to the hash map. However, in practice, running time decreases if the DP tables are sparsely populated. We stress that the sparse DP nevertheless guarantees to find the optimal solution.

5.2.3 Integer Linear Programming

Integer Linear Programs (ILPs) are a classical approach for finding exact solutions of computationally hard problems. We now present an ILP for computing a pairwise unordered tree alignment. Again, let $T_1 = (V_1, E_1), T_2 = (V_2, E_2)$ be the input trees with $V_1 \cap V_2 = \emptyset$. As the ILP is edge-based, we have to introduce some additional notation: Let $e \in E_i, i \in \{1, 2\}$, be any edge in one of the two given trees. We denote by $\mathcal{D}(e)$ the set of *edges* in the subtree rooted at the head of e , and by $\mathcal{N}(e) := E_i \setminus (\{e\} \cup \mathcal{D}(e))$ the non-descendant edges of e . For an edge e , we define $p(e)$ to be the *parent edge*, and $p^*(e) := \{p(e), p(p(e)), \dots\}$ all of its ancestor edges. Finally, $\mathcal{F}(e) := \mathcal{D}(p(e)) \cap \mathcal{N}(e)$ is the “extended family” of e , that is, all descendants of e ’s parent edge, except for e and its descendants.

We start with the ILP without considering the JOIN operation and use the following binary variables: Iff an edge $e' \in (E_1 \cup E_2)$ appears in the aligned subtree, we have $z_{e'} = 1$; iff this edge is aligned to a gap, we have $y_{e'} = 1$. Finally, iff an edge $e \in E_1$ is aligned to an edge $f \in E_2$, we have $x_{\{e,f\}} = 1$. The Constraints (5.11) ensure for each edge that we decide whether this edge is used in the alignment and if, how it is aligned. (5.12) ensure that the subgraphs of T_1 (and T_2) are proper trees. Finally, Inequalities (5.13) ensure that the obtained alignments are consistent: assume an alignment $\langle e, f \rangle$ then we cannot also align a descendant of e with a non-descendant of f and vice versa. The conditional term following the universal quantifier simply avoids redundancy.

$$\max \sum_{\substack{e \in E_1, \\ f \in E_2}} \sigma(e, f) \cdot x_{\{e,f\}} + \sum_{e' \in E_1} \sigma(e', -) \cdot y_{e'} + \sum_{e' \in E_2} \sigma(-, e') \cdot y_{e'} \quad (5.10)$$

$$\text{s.t.} \quad y_e + \sum_{f \in E_{3-i}} x_{\{e,f\}} = z_e \quad \forall i \in \{1, 2\}, e \in E_i \quad (5.11)$$

$$z_{e'} + z_{e''} \leq 1 + z_e \quad \forall i \in \{1, 2\}, e \in E_i, e' \in \mathcal{D}(e), e'' \in \mathcal{F}(e) \quad (5.12)$$

$$x_{\{e,f\}} + x_{\{e',f'\}} \leq 1 \quad \forall i \in \{1, 2\}, e \in E_i, f \in E_{3-i}, \quad (5.13)$$

$$\begin{aligned} & e' \in \mathcal{D}(e), f' \in \mathcal{N}(f), [\text{if } i = 2: f' \notin p^*(f)] \\ & x_{\{e,f\}}, y_{e'}, z_{e'} \in \{0, 1\} \quad \forall e \in E_1, f \in E_2, e' \in (E_1 \cup E_2) \end{aligned} \quad (5.14)$$

Based thereon, we can construct an ILP allowing JOIN operations. Therefore we require additional binary variables $x_{\{e,f\}}^{(i)}$ (with $i \in \{1, 2\}, e \in E_i, f \in E_{3-i}$) which are 1 iff the joined edges $(p(e), e)$ are aligned with f . Technically, we also require $x_{\{e,f\}} = 1$ in such a case. Note that this amount of additional variables is necessary to compose a linear objective function, when the join-costs cannot be computed only based on align- and gap-costs. Furthermore, we introduce binary variables $\phi_{e'}, e' \in (E_1 \cup E_2)$, which are 1 iff the edge e' is used as a parent edge within a join (e.g., $\phi_{p(e)} = 1$ if the former $x_{\{e,f\}}^{(i)}$ variable is 1). We use the shorthands $\sigma^{(1)}(e, f) := \sigma(e + p(e), f) + \sigma_{\text{join}} - \sigma(e, f)$ and $\sigma^{(2)}(e, f) := \sigma(e, f + p(f)) + \sigma_{\text{join}} - \sigma(e, f)$ in the objective function.

$$\max \sum_{\substack{e \in E_1, \\ f \in E_2}} \left(\sigma(e, f) x_{\{e,f\}} + \sum_{i \in \{1, 2\}} \sigma^{(i)}(e, f) x_{\{e,f\}}^{(i)} \right) + \sum_{e' \in E_1} \sigma(e', -) y_{e'} + \sum_{e' \in E_2} \sigma(-, e') y_{e'} \quad (5.15)$$

$$\text{s.t.} \quad y_e + \phi_e + \sum_{f \in V_{3-i}} x_{\{e,f\}} = z_e \quad \forall i \in \{1, 2\}, e \in E_i \quad (5.16)$$

$$z_{e'} + z_{e''} \leq 1 + z_e \quad \forall i \in \{1, 2\}, e \in E_i, e' \in \mathcal{D}(e), e'' \in \mathcal{F}(e) \quad (5.17)$$

$$x_{\{e,f\}} + x_{\{e',f'\}} \leq 1 \quad \forall i \in \{1, 2\}, e \in E_i, f \in E_{3-i}, \quad (5.18)$$

$$e' \in \mathcal{D}(e), f' \in \mathcal{N}(f), [\text{if } i = 2: f' \notin p^*(f)]$$

$$\phi_{e'} + \phi_{e''} \leq 1 \quad \forall e' \in (E_1 \cup E_2), e'' = p(e') \quad (5.19)$$

$$x_{\{e,f\}}^{(1)} + x_{\{e,f\}}^{(2)} \leq x_{\{e,f\}} \quad \forall e \in E_1, f \in E_2 \quad (5.20)$$

$$x_{\{e,f\}} - x_{\{e,f\}}^{(i)} \leq 1 - \phi_{e'} \quad \forall i \in \{1, 2\}, e \in E_i, e' = p(e), f \in E_{3-i} \quad (5.21)$$

$$y_e \leq 1 - \phi_{e'} \quad \forall i \in \{1, 2\}, e \in E_i, e' = p(e) \quad (5.22)$$

$$x_{\{e,f\}}^{(i)} \leq \phi_{e'} \quad \forall i \in \{1, 2\}, e \in E_i, e' = p(e), f \in V_{3-i} \quad (5.23)$$

$$x_{\{e,f\}}^{(i)} + \phi_{f'} \leq 1 \quad \forall i \in \{1, 2\}, e \in E_i, f \in E_{3-i}, f' = p(f) \quad (5.24)$$

$$x_{\{e,f\}}, y_{e'}, z_{e'}, x_{\{e,f\}}^{(i)}, \phi_{e'} \in \{0, 1\} \quad \forall i \in \{1, 2\}, e \in E_1, f \in E_2, e' \in (E_1 \cup E_2) \quad (5.25)$$

Constraints (5.16)–(5.18) are analogous to the former ILP. While (5.19) guarantees that joins are always separated from each other within an input tree, (5.20) ensures that at most one joined alignment may occur for any edge. Inequalities (5.21)–(5.23) make sure that a parent edge e' is only marked as a joined parent iff all its aligned children are joined with e' . Finally, (5.24) guarantees that we do not align two joined edges with each other.

5.3 Comparing Running Times of the Algorithms

Aligning fragmentation trees is computationally hard. In the previous section, we presented three exact algorithms for this problem: a dynamic programming (DP) algorithm, a sparse variant of the DP and an Integer Linear Program (ILP). In this section we evaluate our methods on three different datasets and show that thousands of alignments can be computed in a matter of minutes using DP, even for challenging instances.

Table 5.1: The three datasets used in this study. Fragmentation trees were computed for all compounds. Only non-empty trees were considered for tree alignment. The maximum out-degree of a single tree is denoted by out-degree_{\max} . Number of alignments is given without self-alignments. For all three dataset, the rounded average out-degree_{\max} equals the median out-degree_{\max} .

	<i>Orbitrap</i>	<i>MassBank</i>	<i>Hill</i>
number of compounds	97	370	102
number of non-empty trees	93	343	102
maximum out-degree	7	6	10
average/median out-degree_{\max}	3	2	5
number of alignments	4 278	58 653	5 151

5.3.1 Reference Datasets

To evaluate our work, we use three different test datasets (see Table 5.1). All spectra in the three datasets are tandem mass spectra as explained in Section 2.4.2.

Orbitrap dataset The *Orbitrap* dataset contains 97 compounds, measured on an Orbitrap XL instrument (Thermo Fisher Scientific, Bremen, Germany) with a mass accuracy below 5 ppm [113]. From these, 37 compounds were already used for fragmentation tree evaluation by Rasche *et al.* [112]. Fragmentation was performed using Collision Induced Dissociation (CID) for most compounds. For 26 compounds, High-energy Collision Dissociation (HCD) was used as fragmentation technique.

MassBank dataset The *MassBank* dataset consists of 370 compounds measured on a Waters Q-ToF Premier mass spectrometer. This dataset was downloaded from the MassBank database [55] with accession numbers PR100001 to PR101056. We discarded compounds where the measurement of the unfragmented molecule mass deviated more than 10 ppm from the theoretical mass. Mass accuracy 50 ppm for the analysis was chosen by manual inspection of the data. So, mass accuracy is one order of magnitude worse than for the *Orbitrap* data.

Hill dataset The *Hill* dataset consists of 102 compounds measured on a Micromass Q-ToF, published by Hill *et al.* [52].

For each compound in the three presented datasets, we calculate a hypothetical fragmentation tree from the tandem mass spectra, as described in [7, 112]. Here, the fragmentation model is the same as outlined in Section 4.1.1, but the scoring of the fragmentation graph is different for tandem mass spectra. Further, we assume the molecular formula of the compound to be known. Fragmentation trees were computed using Integer Linear Programming as presented by Rauf *et al.* [114]. Only non-empty trees are considered for tree alignment. Self-alignments are excluded from the analysis (see Table 5.1).

5.3.2 Running Time Comparison

For the alignment of the fragmentation trees, we use a scoring function very similar to the one from Rasche *et al.* [113] that was used to show the applicability of

Table 5.2: Total running times for an all-against-all alignment of the *MassBank* and *Orbitrap* datasets for the three presented algorithms.

	# trees	classical DP	sparse DP	ILP
<i>MassBank</i>	343	4.2 s	1.8 s	9.6 min
<i>Orbitrap</i>	93	5.4 s	0.6 s	14.5 min

fragmentation tree alignments to identify unknown compounds (see Section 5.4). The scoring function is evaluating both pairs of losses and pairs of fragments. For losses l_1, l_2 , we distinguish between size-dependent positive match scores $\sigma(l, l) := 5 + \text{number of non-hydrogen atoms}$ and size-dependent negative mismatch scores $\sigma(l_1, l_2) := -5 - \text{number of different non-hydrogen atoms}$. For fragments f_1, f_2 , we use size-dependent positive match scores $\sigma(f, f) := 5 + \text{number of non-hydrogen atoms}$ and size-independent negative mismatch scores $\sigma(f_1, f_2) := -3$. We allow insertion/deletions, as well as joining two subsequent losses, both without penalty. The idea behind this *ad hoc* scoring is to reward or penalize large losses stronger than small losses, whereas non-matching fragments are penalized independent of size.

We implemented the DP algorithms in Java 1.6. For the sparse DP, we used lazy arrays to store the DP tables. We solved the ILP via branch and cut using CPLEX 12.1 in its default settings. Computation was done on two different but comparable computers, namely on a quad-core 2.2 GHz AMD Opteron processor with 5 GB of main memory for the DP algorithms, and on a quad-core Intel Xeon E5520 with 2.27 GHz in 32-bit mode for the ILP, using 2 GB RAM per job. For the DP algorithms, we repeated computations five times, reporting the minimum running time for each instance.

For the *Orbitrap* and the *MassBank* dataset, we found that for over 98 % of the instances the running time was in the range of microseconds for both DP algorithms. For these datasets, we only evaluate total running times for all alignments (see Table 5.2). For *MassBank*, the classical DP finished in 4.2 s for an all-against-all alignment of 343 trees, whereas sparse DP only required 1.8 s. For *Orbitrap*, the classical DP finished in 5.4 s for the all-against-all alignment of 93 trees, whereas sparse DP required 0.6 s, a nine-fold speed-up. In contrast, the ILP needed 9.6 min for all alignments in the *MassBank* datasets and 14.5 min for all alignments in the *Orbitrap* dataset.

The *Hill* dataset contains trees with much higher maximum out-degree, so we performed a more detailed running time analysis. Classical DP required 13.9 min and sparse DP finished in 1.3 min, an eleven-fold speed-up. Running times of the ILP could only be measured without allowing JOIN operations. For 1241 instances, computations run into the memory limitation of 2 GB. For the remaining alignments, the ILP finished in 11.24 h. Hence, we excluded the ILP from our detailed analysis. To get an overview of differences in the running times between hard and easy alignments, we sorted the instances by their running times in increasing order. This was done separately for each algorithm (see Figure 5.6 (left) and Table 5.3). For both algorithms, we found that the 99 % fastest alignments need nearly as much computing time as the remaining 1 % slowest alignments (see Figure 5.7). We further sorted all instances by the running time of the classical DP (see Figure 5.6 (right)). We found that for every instance, sparse DP requires less time than the classical DP.

Table 5.3: Running times for the *Hill* dataset. We report running times in seconds and as fractions of the total running time for all instances (5151 alignments). We also report running times for the 90 % and 99 % fastest and for the 1 % slowest alignments. For both algorithms, instances were sorted separately.

	all	90 % fastest	99 % fastest	1 % slowest
dynamic programming	833.3 s	133.5 s (16.0 %)	437.9 s (52.6 %)	395.4 s (47.4 %)
sparse dynamic programming	75.3 s	13.9 s (18.5 %)	33.9 s (45.0 %)	41.4 s (55.0 %)
speed up	11-fold	10-fold	13-fold	10-fold

Evaluation of our methods on the three different datasets showed that thousands of alignments can be computed in a matter of minutes using DP, even for challenging instances. We find that the sparse DP approach dominates the classical DP, resulting in an eleven-fold speed-up for the *Hill* dataset (see Figure 5.7). The ILP is usually clearly outperformed by both DP approaches; nevertheless, it has the potential to solve those instances that are “hard” for DP-based algorithms. In fact, a large fraction of the total running time stems from a few “hard” alignments which, in turn, correspond to a few trees in the dataset that are large and, in particular, have high out-degrees. For larger datasets, we expect that the running time spend on computing the 99 % fastest alignments will be significantly smaller than the running time spend on the 1 % slowest alignments.

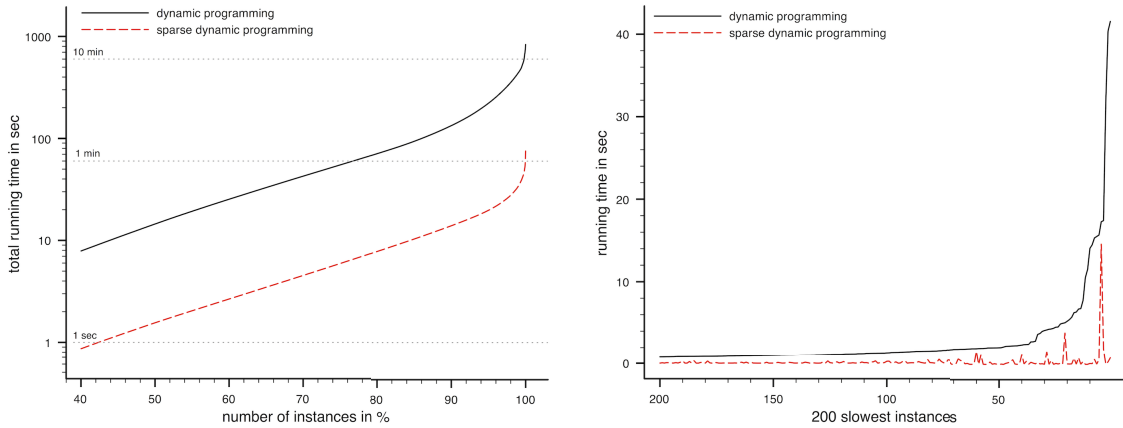


Figure 5.6: Running times for the *Hill* dataset with 5151 individual alignments. Left: Total running times when instances are sorted by individual running times. For any fraction $x\%$, we calculate the total running time of the $x\%$ instances for which the alignment was computed faster than for any of the remaining instances. For example, at 50 % one can find the running time that was needed to compute the 50 % fastest instances. For each algorithm, instances were sorted separately. Note the logarithmic y-axis. Right: Individual running times for the 200 slowest instances of the classical DP algorithm. Instances are sorted by their running time for the classical DP algorithm. One can see that running times of the classical DP are outperformed by that of the sparse DP.

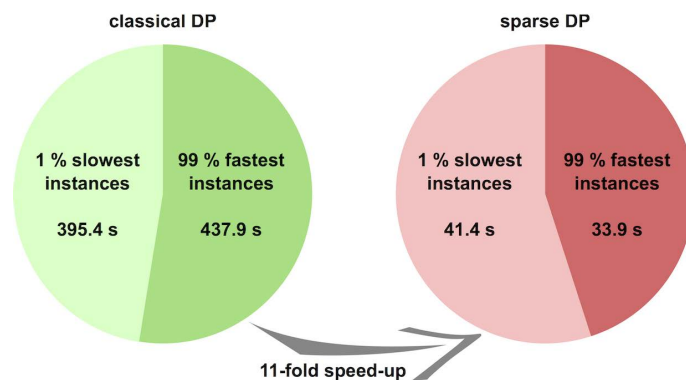


Figure 5.7: Running times for the 99 % fastest and for the 1 % slowest instances of the *Hill* dataset for both DP algorithms. For both algorithms, instances were sorted separately.

5.4 Application of Fragmentation Tree Alignments: Clustering Similar Compounds

Aligning fragmentation trees is used to derive useful information beyond the molecular formula of an unknown compound by identifying fragmentation cascades that similarly occur in already known compounds.

Rasche *et al.* [113] proposed three applications of fragmentation tree alignment: (1) clustering a set of known and/or unknown compounds based on the similarity scores to predict structural properties and/or compound classes; (2) correlating the similarity score of two fragmentation trees with the Tanimoto structural similarity score of the corresponding compounds to assess the quality of fragmentation tree alignment; (3) searching for structurally similar compounds in a spectral library using *FT-BLAST* allowing for significance estimation of the hits using a decoy database strategy.

Here, we again demonstrate how to use clustering to conclude structural information for a set of unknown compounds. The results of this section have already been presented by Rasche in his thesis [111]. We use a set of known reference compounds to demonstrate that clusters based on fragmentation pattern similarity show a good agreement with known compound classes. We then apply the method to unknown metabolites from Icelandic poppy and derive compound class and structural information to simplify downstream NMR analysis.

The results presented in this section are based on the older version of the DP algorithm [113]. In this version, only one child node is allowed to be joined with its parent, all other children are discarded.

5.4.1 MS Datasets

We demonstrate the clustering approach on a reference dataset of knowns and a real-world dataset of unknowns. First, we show that similar compounds (belonging to the same compound class) from the *Orbitrap* dataset (see Section 5.3.1 and Table 5.1) cluster together. The *Orbitrap* dataset mainly contains zeatins, amino acids, glucosinolates, sugars and benzopyrans. As a real-world example of using our method we analyze several extracts from Icelandic poppy (*P. nudicaule*) in an Orbitrap mass spectrometer and apply clustering

to a combined dataset of these unknown compounds and the reference compounds from the *Orbitrap* dataset.

Poppy dataset Surface extracts of *P. nudicaule* were made using methanol: 1 % acetic acid 2:1 mixture. The following organs of the plant were processed in different samples: petals, stamen with and without base, and stem. The extracts were directly infused using a Nanomate Triversa system (Advion, Ithaca, NY) on a Nanomate nanoelectrospray chip and analyzed on an Orbitrap XL (Thermo Fisher Scientific, Bremen, Germany). The instrument operated at 100 000 resolution and settings for tandem mass spectra acquisition as for the *Orbitrap* dataset. Measurements were conducted in both positive and negative mode using several collision energies. Precursor ions were manually selected based on ion intensities and expected masses obtained from literature, and HCD-fragmented. The data contained 489 non-empty fragmentation spectra of 89 potential compounds.

Different from the datasets described in Section 5.3.1, here the molecular formulas of the compounds are unknown. To determine the molecular formulas we use both isotope pattern analysis and fragmentation trees as described by Rasche *et al.* [112]. Unfortunately, isotope patterns were often of insufficient quality: In many cases, only the monoisotopic and the $M + 1$ isotope could be detected. To this end, we conservatively selected 29 poppy compounds where fragmentation tree analysis and isotope pattern analysis agreed upon the molecular formula of the unknown: For these compounds, the best ranked molecular formula of the combined analysis is among the top five molecular formulas of the isotope pattern analysis, and among the top five molecular formulas of the fragmentation pattern analysis. For each of the 29 poppy compounds, we calculate a fragmentation tree as described by Rasche *et al.* [112].

5.4.2 Scoring Alignments

To derive useful information from fragmentation tree alignments, similarity between two trees can be measured by comparing losses, comparing fragments, or even comparing both. In this evaluation, we base our fragmentation tree alignment on comparing both, hence we need a scoring function to evaluate pairs of losses, as well as pairs of fragments (see Table 5.4). We distinguish three main cases for two losses l_1 and l_2 :

- For a *match* $l_1 = l_2$, we assign a size-dependent positive score: $\sigma(l, l) := 5 + \#atoms$ where $\#atoms$ is the number of non-hydrogen atoms in the loss l (that is all carbon atoms and heteroatoms).
- For a *mismatch* $l_1 \neq l_2$, we assign a size-dependent negative score $\sigma(l_1, l_2) := -2 - 0.5\#diff$ where $\#diff$ is the number of non-hydrogen atoms in the symmetric difference between the two losses.
- For an *insertion/deletion* (indel) where either $l_1 = -$ or $l_2 = -$ is a gap symbol, we set $\sigma(l_1, -) = \sigma(-, l_2) = 0$, as deleting nodes from the alignment implicitly reduces the score that can be reached.
- For *joining* two subsequent losses, we set $\sigma_{join} = 0$, as joining two subsequent losses implicitly reduces the score that can be reached by the alignment.

Scoring of two fragments f_1 and f_2 is somewhat similar:

Table 5.4: Scoring neutral losses and fragments.

	event	score
losses	basic match score	+5
	modification for each non-hydrogen atom	+1
	basic mismatch score	-2
	modification for each non-hydrogen atom	-0.5
fragments	basic match score	+5
	modification for each non-hydrogen atom	+1
	basic mismatch score	-3
	modification for each non-hydrogen atom	± 0
	insertion/deletion score	± 0
	joining two subsequent losses	± 0

- For a *match* $f_1 = f_2$, we assign a size-dependent positive score $\sigma(f, f) := 5 + \#atoms$ where $\#atoms$ is the number of non-hydrogen atoms in the fragment f (that is all carbon atoms and heteroatoms).
- For a *mismatch* $f_1 \neq f_2$, we assign a size-independent negative score $\sigma(f_1, f_2) := -3$. In this way, we allow for matching losses even when the corresponding fragments show no similarity.

Some compounds in the *Orbitrap* dataset are isotopically labeled with deuterium. When comparing molecular formulas of losses or fragments in the alignment, we treat deuterium as hydrogen. As an example, aligning losses H_2O and HDO would receive a score of +6.

5.4.3 Normalization of Scores and Fingerprinting

Since the score of an alignment is highly dependent on the size of the trees, alignment scores have to be normalized. In the extreme case of a fragmentation tree with only one node (the parent molecule), the alignment score is zero against *all* other trees. To this end, we normalize by the score a *perfect match* would obtain. Since we do local alignments, a perfect match means that one tree is a subtree of the other tree. The same score is obtained by aligning this subtree with itself, $S(T_i, T_i)$. We normalize the score by

$$S_0(T_1, T_2) = \frac{S(T_1, T_2)}{(\min\{S(T_1, T_1), S(T_2, T_2)\})^c} \quad (5.26)$$

where $c \in [0, 1]$ is the normalization parameter. For a total normalization by perfect match score, we use $c = 1$. This normalization favors small trees and discriminates against large trees, since it is much more likely for a very small tree to be a subtree of another tree, than for a medium-size or large tree. In the following, we do not use total normalization, but rather choose $c = \frac{1}{2}$.

When two compounds are structurally similar, they should show comparable fragmentation tree similarities to *any* other compound. To this end, we use the scores of one compound against all others as its *fingerprint*. We compare two compounds by comparing their fingerprints. This can be achieved using any classical methods for

comparing fingerprint vectors, such as Euclidean distance or Pearson correlation. We use the well-known Pearson product-moment correlation coefficient r that measures the linear dependence of two variables $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (5.27)$$

with $-1 \leq r \leq +1$. Here, \bar{X} denotes the mean of X_1, \dots, X_n .

5.4.4 Clustering Fragmentation Trees

Clustering compounds based on fragmentation tree similarity is used to derive compound class and structural information of unknown compounds (see Figure 5.8). A fragmentation tree is computed from the measured spectrum for each compound in the dataset of unknowns. These trees are aligned in an all-against-all manner to a database of fragmentation trees derived from reference compounds, for example a spectral library. Both known and unknown compounds are clustered together based on the resulting similarity scores. Similar compounds that belong to the same compound class and/or share common substructures cluster together. The compound class and structural information of the unknown compound can be concluded from the cluster into which it falls. Even if no reference compounds are available, a clustering of the unknown measurements gives a first overview of the dataset. An experienced experimenter may even spot the compounds of interest for his study from such a clustering.

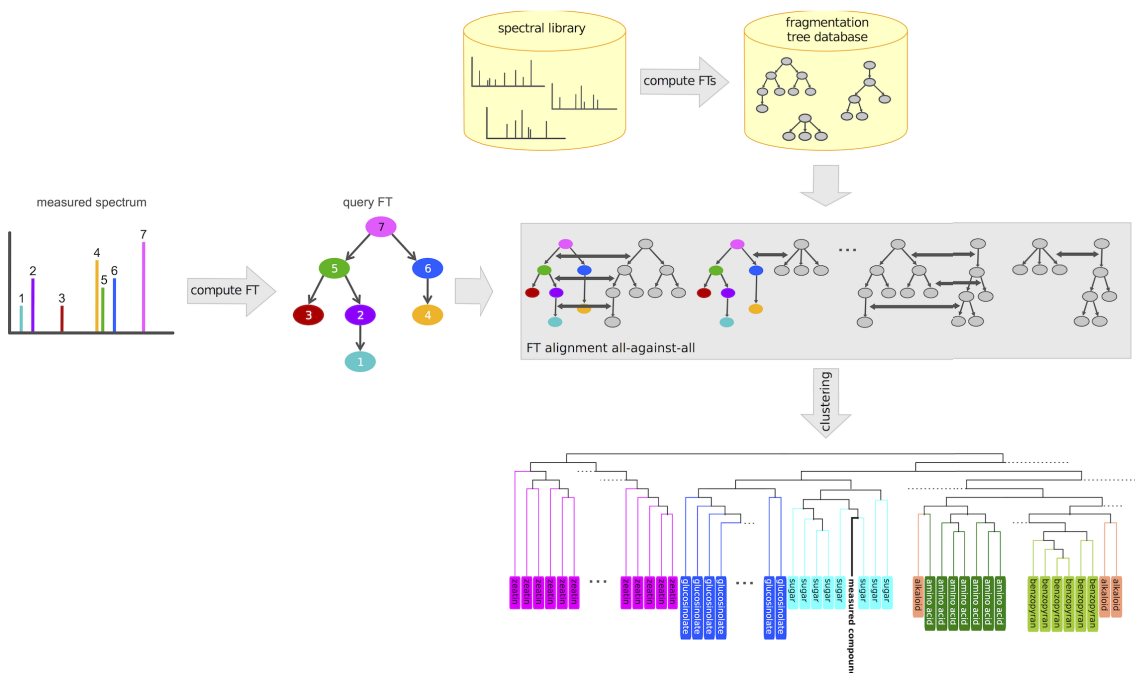


Figure 5.8: Fragmentation tree alignment for compound classification. A fragmentation tree is computed from the measured spectrum. The tree is aligned to a database of fragmentation trees in an all-against-all manner. The compounds are clustered based on the resulting similarity scores. Similar compounds (belonging to the same compound class) cluster together. The class of the unknown compound can be concluded from the cluster into which it falls.

First, we perform all-against-all pairwise alignments for the *Orbitrap* dataset. We normalize scores and compute *fingerprints* of the compounds as described in Section 5.4.3. This results in a matrix of pairwise similarities. To this matrix, we apply hierarchical clustering, more precisely, UPGMA (Unweighted Pair Group Method with Arithmetic Mean) agglomerative clustering [141] using EPoS [42]. Although hierarchical clustering is probably not the best-suited method for clustering compounds based on fragmentation tree similarity, we have chosen this method as it is well-known from other applications [22].

It is understood that for fragmentation trees with few losses, clustering results will become somewhat arbitrary: In the extreme case of a single neutral loss, similarity or dissimilarity to any other fragmentation tree can easily be spurious. To this end, we limit calculations for the *Orbitrap* dataset to fragmentation trees with three and more losses (that is 77 compounds). We believe that this is not a shortcoming of our method, but rather the problem that certain compounds do not “fragment sufficiently” under tandem MS, resulting in mostly uninformative fragmentation spectra. This problem may be overcome by using multiple MS.

Second, we cluster the unknowns from the *Poppy* dataset together with the reference measurements from the *Orbitrap* dataset. Again, we perform all-against-all pairwise alignments for all compounds from the combined dataset and use normalized fingerprint similarities for hierarchical clustering. Here, we use all fragmentation trees with at least one loss to include as many reference compounds as possible (that is 93 compounds from the *Orbitrap* dataset and 29 compounds from the *Poppy* dataset).

5.4.5 Clustering Results of the Reference Dataset

We first analyze the reference dataset to show that similar compounds (belonging to the same compound class) cluster together. We discard 20 compounds from the *Orbitrap* dataset as the resulting fragmentation trees show less than three losses. The *Orbitrap* dataset contains mostly zeatins (21 with at least three losses), glucosinolates (14), benzopyrans (11), sugars (9), and amino acids (9). The detailed clustering and the clustering with collapsed mostly-homogeneous clusters is depicted in Figure 5.9. We observe that clusters are very homogeneous: There is a perfect glucosinolate cluster containing all 14 glucosinolates, a perfect zeatin cluster containing all 21 zeatins, and an almost perfect sugar cluster containing all nine sugars, plus one anthocyanin and one carboxylic acid. Furthermore, there is an almost perfect amino acid cluster containing seven of the nine amino acids plus one alkaloid. Similarly, there is a perfect benzopyran cluster containing six of the eleven benzopyrans.

These results show the capability of the method to differentiate compound classes. Large compound classes form almost perfectly separated clusters. Smaller compound classes are distributed among several clusters, but clusters contain few outliers. We apply hierarchical clustering as a proof-of-concept to demonstrate clustering results. Better results can possibly be achieved by other clustering methods and supervised Machine Learning. Nevertheless, our results indicate how to deduce the compound class of an unknown if a reasonable number of knowns are clustered simultaneously.

5.4.6 Identifying Unknowns from a Biological Sample

We cluster unknowns from Icelandic poppy (*P. nudicaule*) together with the *Orbitrap* dataset (see Figure 5.10). Eight compounds in the sample were identified by manual

analysis of the spectra: Arginine (175 Da), glutamine (147 Da), quercetin (301 Da) and a hexose (179 Da), as well as four alkaloids, namely, reticuline (330 Da), corytuberine (328 Da), and two hydrogenated and hydroxylated palmatines (370 and 386 Da).

All manually identified unknowns are grouped into their respective cluster. Even though our reference dataset only contains few alkaloids, they cluster together with the four manual identified alkaloids, reticuline, corytuberine, and the two palmatine derivatives, as well as one other unknown (400 Da). This compound probably is also an alkaloid, but since it is located at the border of the cluster, more reference alkaloids are required for a reliable classification. Quercetin (301 Da) is neighboring its respective reference from the *Orbitrap* dataset in a benzopyran cluster. Arginine (175 Da) and glutamine (147 Da)

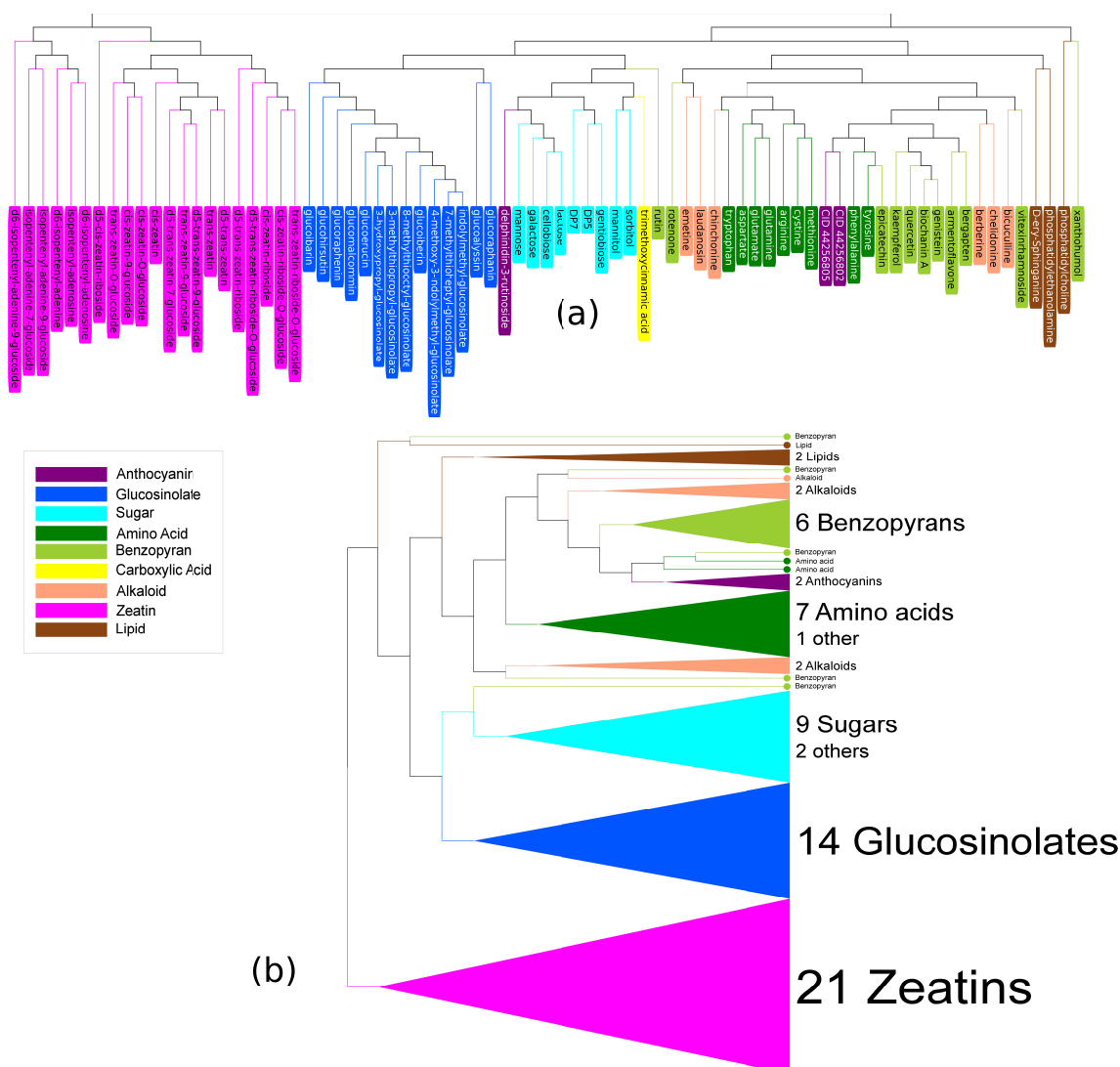


Figure 5.9: (a) Hierarchical clustering based on fragmentation tree fingerprint similarities of the *Orbitrap* dataset (compounds with three and more losses, $N = 77$). (b) For better visualization, we have collapsed (mostly) homogeneous clusters. Glucosinolates (blue) and zeatins (magenta) form perfect clusters, all sugars (cyan) form a cluster together with two other compounds, and large groups of amino acids (green) and benzopyrans (light green) form almost perfect clusters.

are both grouped into amino acid clusters. Since the unknown at 229 Da falls into the amino acid cluster, too, we consider it at least strongly related with amino acids. The unknown hexose (179 Da) is most likely glucose, which was not in our reference, but is grouped together with many other sugars. The 277 Da molecule is probably a sugar, too, or contains a sugar moiety. With the limited reference data, it is not possible to assign a group to the 438 Da and 537 Da compounds, which form a separated cluster. Manual interpretation failed to identify the compounds. We may assume that they are neither related to zeatins nor to glucosinolates. Actually, no unknown falls into the well-separated zeatin and glucosinolate clusters. Additionally, our analysis correctly shows that a contamination with mass 338 Da, measured during a blank column run, is similar to the lipids. Database search and manual validation identified it as erucamide (PubChem CID 5365371), an additive originating from the plastic ware used for sample collection.

Results from the clustering analysis can be seen as strong hints towards a compound class. This can point towards unknowns of interest and simplify the downstream analysis, e.g. using NMR. The analysis of unknowns will become more powerful as more reference compounds become available.

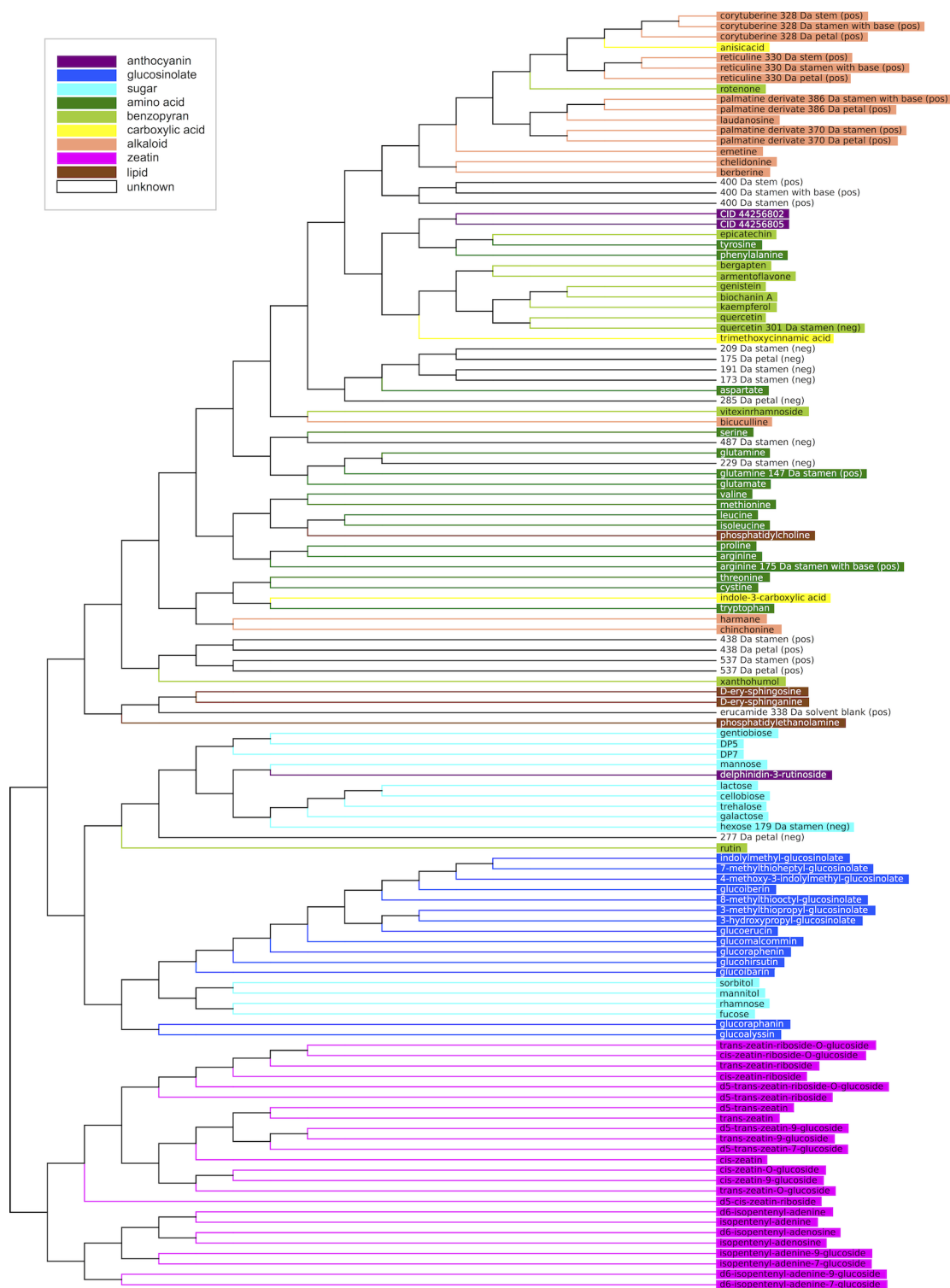


Figure 5.10: Combined clustering of the *Poppy* and the *Orbitrap* datasets, only non-empty trees were considered. Colored compounds are known references from the *Orbitrap* or manually identified compounds from the *Poppy* dataset. Many unknown compounds form a cluster together with several alkaloids (top of the figure). Other unknowns end up in amino acid or sugar clusters. The poppy sample most likely contained no glucosinolates and zeatins, as no unknowns can be found among these clusters.

6 Conclusion

In this thesis, we have presented novel computational methods for the automated analysis of high-accuracy fragmentation mass spectra of small molecules. Previous approaches use either spectral libraries or the more comprehensive molecular structure databases for compound identification. To overcome the limits of the “known universe of organic chemistry”, a true *de novo* analysis of fragmentation data is required. This work presents methods that target “true unknowns” which are contained in neither spectral nor molecular structure databases. Thus, our methods are capable of supporting the explorative character of metabolomics studies.

We presented the *de novo* interpretation of high resolution EI fragmentation data based on fragmentation tree construction. Our method is independent of existing library knowledge and, besides a list of common losses, it does not use any chemical expert knowledge. Our method can identify the molecular ion peak and molecular formula of a metabolite if the molecular ion is present in the spectrum, and further explain relevant fragmentation reactions. Many available methods for analyzing fragmentation spectra of metabolites are rule-based. Completely unknown compounds may not necessarily follow these known rules for classification or fragmentation. In contrast, our method is a rule-free approach and is not limited to known compound classes. Fragmentation trees are constructed from high-accuracy EI spectra by automated signal extraction and hypothesis-driven evaluation. As we did not train the parameters of our combinatorial optimization method using the data, it is in principle applicable to any compound class.

Given the fragmentation data of an unknown small molecule, our approach proceeds in two steps: First, we try to pick the molecular ion peak and derive a molecular formula. This is even possible if the molecular ion is hidden under contaminants with higher masses or is low in abundance. Such difficulties are frequently observed, because EI is a hard ionization method that often produces highly abundant generic fragment ions, but low abundance or no ions of larger fragments and the unfragmented molecular ion [79]. Our method does not work for mass spectra where the molecular ion is absent, which is the case in about 30 % of spectra [89]. For optimal performance, molecular ion peak and formula identification are performed simultaneously. Second, we compute a fragmentation tree that offers a hypothetical interpretation of the experimental data. These trees explain relevant fragmentation reactions and assign molecular formulas to fragments.

Several parameters limit GC-MS analysis, but major disadvantages occur when the analyte has a high boiling point or breaks down at high temperatures. Compounds with these problems need to be derivatized to reduce their boiling point and protect them against thermal degradation. On the other hand, selective derivatization helps with the detection of functional groups [44]. For metabolomics, derivatization with MSTFA [44] or PFBHA [80] are routine in GC-MS analyses. Our method is capable of dealing with such derivatized compounds.

We evaluated the capabilities of our method on different levels. First, we used measurements from 50 derivatized and underivatized metabolites to evaluate the identification of molecular ion peaks and molecular formulas. The molecular ion was

correctly identified in 88 % of cases, and in 78 % of cases the molecular formula was also correctly assigned. We showed that the molecular ion and formula identification even works on challenging compounds with a molecular ion peak of low intensity. We compared our method to the well established algorithm from Scott [132] and found that the results of the two methods complement each other. While Scott’s algorithm only estimates nominal masses, we obtain the exact mass of the molecular ion if its peak is present in the spectrum. Therefore, our method *does* require high mass accuracy of the measurements, which is further necessary to limit possible molecular formulas.

Further, we used these 50 reference compounds and another 22 compounds with fragmentation pathways that are well annotated in the literature to evaluate fragmentation tree quality. We do not claim the pathways in the trees to be “true” fragmentation processes. We found, however, that the trees correspond very well to published mechanisms and agree in their general information with expert knowledge of EI fragmentation patterns. All important fragmentation reactions of the different compound classes were identified. For the 22 annotated compounds, we found that fragmentation trees explain the origin of the ions found in the mass spectra in accordance with the literature. No peak was annotated with an incorrect fragment formula and 79 % of the fragmentation processes were correctly reconstructed. For the remaining 50 reference compounds, we checked the tree quality by expert evaluation and discussed selected examples taken from the literature. Our method allowed the assignment of specific relevant fragments and fragmentation pathways even in the most complex EI spectra in our dataset.

In addition, we evaluated our method against MetFrag for fragmentation prediction and found an agreement for 75 % of the peak annotations. Rather than applying both methods independently (as done here for comparison), information obtained by our method, such as fragment formulas, can be used to simplify *in silico* fragmentation and presumably improve its results.

To further process fragmentation trees, Rasche *et al.* [113] proposed several workflows using fragmentation tree alignment. Performing these workflows on a large dataset requires tree alignments to be executed extremely fast. We have presented three exact algorithms for the alignment of fragmentation trees: a dynamic programming (DP) algorithm, a sparse variant of the DP, and an Integer Linear Program (ILP). Evaluation of our methods on three different datasets showed that thousands of alignments can be computed in a matter of minutes using DP, even for challenging instances. We found that the sparse DP approach dominates the classical DP, resulting in an eleven-fold speed-up for one dataset. ILPs have an excellent record of providing fast algorithms for NP-hard problems. Thus, it was rather unexpected that, for the problem discussed here, the ILP was usually clearly outperformed by both DP approaches; nevertheless, it has the potential to solve those instances that are “hard” for DP-based algorithms.

When larger datasets become available, we expect the total running time of an all-against-all alignment to increase more than quadratically with dataset size: We have shown that a large fraction of the total running time stems from a few “hard” alignments which, in turn, correspond to a few trees in the dataset that are large and, in particular, have high out-degrees. We conjecture that for larger datasets, the running time spend on computing the 99 % fastest alignments will be significantly smaller than the running time spend on the 1 % slowest alignments. Here, even faster methods for computing fragmentation tree alignments are sought.

Because it clusters compounds based solely on their fragmentation patterns, (which we show can be computed automatically) fragmentation tree alignment allows for an automated classification of unknown compounds into compound classes. Our results indicate how to deduce the compound class of an unknown when a reasonable number of knowns are clustered simultaneously. Thus, large-scale compound screens can easily be searched for compounds of interest, limiting work spent on ubiquitous “uninteresting” molecules. The information deduced from fragmentation trees and fragmentation tree alignments may also simplify downstream NMR analysis.

6.1 Future Work

Our method does not work for mass spectra where the molecular ion is absent, which is the case in about 30 % of spectra [89]. To predict the molecular ion peak and molecular formula for those compounds, we want to follow the line of thought described by Scott [132]: First, we want to reconstruct the molecular formula of one or more large fragments as reference fragments by computing a fragmentation forest. Then, we can use machine learning to predict the molecular formula of the “hidden loss” and recompute the fragmentation tree using the (hypothetically) identified molecular formula as root.

Although our evaluation indicated that fragmentation trees are already of high quality, better scoring functions could improve the results. By testing numerous combinations of the different scoring parameters, it would probably be possible to find better scoring functions. For the low amount of data at hand, such parameter optimization would have led to “overfitting” to the available data, so we set the parameters to default values or values that were chosen ad hoc. Given larger datasets, new scoring functions based on statistical considerations can be derived, as has been already done for tandem MS data [29], such as prior probabilities for all losses. Further, we want to include an important piece of information present in EI mass spectra but currently ignored by our method: namely, isotopic patterns of the molecule and its fragments. Using this information for both the intact molecule and its fragments is a straightforward idea, possibly first proposed back in 1988 by Tenhosaari [152].

EI spectra contain many more peaks than fragmentation spectra from CID. Faster methods for computing fragmentation trees allow this information to be used for the identification of the molecular ion peak and molecular formula, and to avoid the two-step approach for analysis of EI spectra. In combination with improved scoring, this may help to increase the number of correctly predicted molecular formulas.

We expect that better-quality fragmentation trees will improve tree alignment results and further downstream analysis.

As in the case of computing fragmentation trees, we believe that a better scoring of fragmentation tree alignments will improve the quality of these alignments. In our evaluations, we have used a scoring function similar to the one by Rasche *et al.* [113]. Both scoring functions lack any statistical explanation and should be refined in the future using, for example, log-odds or log-likelihood scores. Parameters can be learned using experimental data, since for tandem MS we now have larger reference datasets at hand.

Besides finding a better scoring function, we can also further modify the alignment problem itself. Fragmentation tree alignment already accounts for the combination of consecutive edges, but this is currently limited to joining exactly two edges to avoid a combinatorial explosion. Allowing multiple joined losses may further improve the

quality of fragmentation tree alignments. In addition, we found that for some consecutive fragmentation steps, the respective ions do not allow the correct fragmentation order to be determined solely from the fragmentation MS data. In the examined dataset, these configurations occurred in 86 % of the compounds. In the future, both fragmentation possibilities should be considered in the fragmentation tree alignment.

Aligning fragmentation trees is computationally hard; specifically, NP-hard. We have presented a method to compute thousands of alignments in a matter of minutes. However, even faster methods for computing fragmentation tree alignments are sought for several reasons: The most obvious reason is the increasing size of fragmentation tree libraries to search against. Further, we have shown that a large fraction of the total running time stems from a few trees in the dataset that are large and, in particular, have high out-degrees. In addition, for more accurate estimation of the significance of a hit from FT-BLAST, the size of the decoy database has to be increased. Finally, the above mentioned modifications of the alignment problem can lead to even more demanding computational problems.

There are already several ideas to speed up computation. The ILP approach should be pursued, even though it is (somewhat unexpectedly) outperformed by both DP approaches. We plan to evaluate whether the ILP is capable of solving the "hard" instances faster than a DP-based approach, as its running time is not directly dependent on the out-degree of the trees. In addition, we can use classical Lagrangian Relaxation [90] to speed up computation, but this may require formulation of a different ILP.

Even though small trees with low out-degree seem to be less interesting since they often belong to small known compounds (below 300 Da), we believe that we can also speed up alignments for such compounds: this may be achieved using some preprocessing for small trees with, e.g., less than four losses.

Another interesting question is whether polynomial-time methods for tree alignment of unordered trees, such as that used for calculating the constrained tree edit distance [168], can be used for aligning fragmentation trees: whereas the restrictions imposed by Zhang [168] have no sensible interpretation in the context of fragmentation trees, the quality of results may still be sufficient for certain applications.

Originally, fragmentation tree alignment was targeted towards tandem mass spectra. We want to adapt the method for the alignment of trees computed from EI mass spectra. To this end, it might be necessary to modify the scoring, again using a statistical model as mentioned above. As EI mass spectra usually contain more peaks than tandem mass spectra, the resulting trees usually have higher maximum out-degree (stronger branching). As the running time of our method for aligning fragmentation trees depends exponentially on the out-degree of nodes, this is another reason to look for faster alignment algorithms.

Various applications of fragmentation trees and fragmentation tree alignments are possible. We have in mind a pipeline which suggests only a few molecular structures and thus can greatly reduce manual analysis time. Rasche *et al.* [113] recently introduced *FT-BLAST*, a method to find similar, but not necessarily identical, compounds by aligning fragmentation trees. Consensus substructures of these hits may be key structural elements of the unknown compound. These can be used within molecular isomer generators to enumerate all structural isomers containing these substructures [31]. First steps have been made towards an automated analysis of the *FT-BLAST* hit lists involving searching for characteristic substructures in these lists [93], but the results are currently not chemically sound.

Fragmentation tree similarity can be further used for the *de novo* reconstruction of networks from metabolite mass spectrometry data. In 2006, Breitling *et al.* [10] reconstructed networks by inferring accurate mass differences between measured metabolites based on high-resolution single-stage MS data. These mass differences give evidence of biochemical transformations between the metabolites and allow the reconstruction of a network. Watrous *et al.* [161] determined structural similarity between metabolites using spectral alignments and connected structurally similar metabolites to reconstruct a network. We propose fragmentation tree similarity as a means for deciding whether two metabolites are connected. Network reconstruction can be, for example, applied to drug degradation data to identify true drug degradation products.

The fragmentation tree concept started back in 2008, and since it is still relatively young, it is difficult to predict its future development in the years to come. Tackling the identification of unknown metabolites using a *de novo* approach based on combinatorial optimization seems to be promising. We hope, that fragmentation trees and their downstream applications, such as tree comparison, network reconstruction, and whatever new ideas will follow, help to support the explorative character of metabolomics studies. It is, however, indisputable that the “incredibly fast and accurate” analysis using computers will always only complement the expertise of human beings. That is why “the marriage of the two is a force beyond calculation.”

Bibliography

- [1] R. M. Acheson, R. T. Aplin and R. G. Bolton. Electron impact induced alkyl-group fragmentation on the acridine nucleus. *Org Mass Spectrom*, 12(8):518–530, 1977. 33, 39, 91
- [2] J. S. Allen. An improved electron multiplier particle counter. *Rev Sci Instrum*, 18(10):739–749, 1947. 7
- [3] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof verification and the hardness of approximation problems. *J ACM*, 45(3):501–555, 1998. 48
- [4] M. Baker. Metabolomics: From small molecules to big ideas. *Nat Methods*, 8:117–121, 2011. 7
- [5] A. Björklund, T. Husfeldt, P. Kaski and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. of ACM Symposium on Theory of Computing (STOC 2007)*, pages 67–74. ACM press, New York, 2007. 30, 55
- [6] S. Böcker and Zs. Lipták. A fast and simple algorithm for the Money Changing Problem. *Algorithmica*, 48(4):413–432, 2007. 17
- [7] S. Böcker and F. Rasche. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinformatics*, 24:I49–I55, 2008. Proc. of *European Conference on Computational Biology (ECCB 2008)*. 18, 22, 25, 26, 27, 29, 31, 60, 75
- [8] S. Böcker, Zs. Lipták, M. Martin, A. Pervukhin and H. Sudek. DECOMP—from interpreting mass spectrometry peaks to solving the Money Changing Problem. *Bioinformatics*, 24(4):591–593, 2008. 17
- [9] S. Böcker, M. Letzel, Zs. Lipták and A. Pervukhin. SIRIUS: Decomposing isotope patterns for metabolite identification. *Bioinformatics*, 25(2):218–224, 2009. 18
- [10] R. Breitling, S. Ritchie, D. Goodenowe, M. L. Stewart and M. P. Barrett. Ab initio prediction of metabolic networks using Fourier transform mass spectrometry data. *Metabolomics*, 2(3):155–164, 2006. 75
- [11] A. W. T. Bristow, K. S. Webb, A. T. Lubben and J. Halket. Reproducible production tandem mass spectra on various liquid chromatography/mass spectrometry instruments for the development of spectral libraries. *Rapid Commun Mass Spectrom*, 18(13):1447–1454, 2004. 13
- [12] K. L. Brown and G. W. Tautfest. Faraday-cup monitors for high-energy electron beams. *Rev Sci Instrum*, 27(9):696–702, 1956. 7

- [13] P. Brown. Kinetic studies in mass spectrometry - IX: Competing [M-NO₂] and [M-NO] reactions in substituted nitrobenzenes. Approximate activation energies from ionization and appearance potentials. *Org Mass Spectrom*, 4(S1):553–554, 1970. 42
- [14] D. Bylund, R. Danielsson, G. Malmquist and K. E. Markides. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography-mass spectrometry data. *J Chromatogr A*, 961:237–244, 2002. 15
- [15] T. Cajka, J. Hajslova, O. Lacina, K. Mastovska and S. J. Lehotay. Rapid analysis of multiple pesticide residues in fruit-based baby food using programmed temperature vaporiser injection-low-pressure gas chromatography-high-resolution time-of-flight mass spectrometry. *J Chromatogr A*, 1186(1-2):281–294, 2008. 11, 16
- [16] E. Champarnaud and C. Hopley. Evaluation of the comparability of spectra generated using a tuning point protocol on twelve electrospray ionisation tandem-in-space mass spectrometers. *Rapid Commun Mass Spectrom*, 25(8):1001–1007, 2011. 13, 17
- [17] J. Claesen, P. Dittwald, T. Burzykowski and D. Valkenburg. An efficient method to calculate the aggregated isotopic distribution and exact center-masses. *J Am Soc Mass Spectrom*, 23(4):753–63, 2012. 18
- [18] Q. Cui, I. A. Lewis, A. D. Hegeman, M. E. Anderson, J. Li, C. F. Schulte, W. M. Westler, H. R. Eghbalian, M. R. Sussman, and J. L. Markley. Metabolite identification via the Madison Metabolomics Consortium Database. *Nat Biotechnol*, 26(2):162–164, 2008. 7
- [19] W. Danikiewicz, K. Wojciechowski, R. H. Fokkens and N. M. M. Nibbering. Electron impact-induced fragmentation of 2,1-benzisothiazoline 2,2-dioxide. *Org Mass Spectrom*, 28:853–859, 1993. 33, 39, 91
- [20] W. Demuth, M. Karlovits and K. Varmuza. Spectral similarity versus structural similarity: Mass spectrometry. *Anal Chim Acta*, 516(1-2):75–85, 2004. 17
- [21] K. Dettmer, P. A. Aronov and B. D. Hammock. Mass spectrometry-based metabolomics. *Mass Spectrom Rev*, 26(1):51–78, 2007. 6, 12
- [22] P. D’haeseleer. How does gene expression clustering work? *Nat Biotechnol*, 23(12):1499–1501, 2005. 67
- [23] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, 4th edition, 2010. 5, 13
- [24] F. Dieterle, B. Riefke, G. Schlotterbeck, A. Ross, H. Senn and A. Amberg. NMR and MS methods for metabolomics. *Methods Mol Biol*, 691:385–415, 2011. 2
- [25] R. Dondi, G. Fertin and S. Vialette. Maximum motif problem in vertex-colored graphs. In *Proc. of Symposium on Combinatorial Pattern Matching (CPM 2009)*, volume 5577 of *Lect Notes Comput Sci*, pages 221–235. Springer, Berlin, 2009. 29
- [26] F. L. Dorman, J. J. Whiting, J. W. Cochran and J. Gardea-Torresdey. Gas chromatography. *Anal Chem*, 82(12):4775–4785, 2010. 10

- [27] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3): 195–207, 1972. 29
- [28] K. Dührkop, K. Scheubert and S. Böcker. Molecular formula identification with SIRIUS. *Metabolites*, 3:506–516, 2013. 16
- [29] K. Dührkop, A. Svatoš and S. Böcker. Fragmentation trees reloaded. In preparation, 2013. 73
- [30] I. Eidhammer, K. Flikka, L. Martens and S.-O. Mikalsen. *Computational Methods for Mass Spectrometry Proteomics*. Wiley, 2007. 6
- [31] J.-L. Faulon, D. P. Visco and D. Roe. Enumerating molecules. In K. B. Lipkowitz, R. Larter and T. R. Cundari, editors, *Reviews in Computational Chemistry*, volume 21, chapter 3, pages 209–286. John Wiley & Sons, Inc., 2005. 74
- [32] M. R. Fellows, J. Gramm and R. Niedermeier. On the parameterized intractability of motif search problems. *Combinatorica*, 26(2):141–167, 2006. 29
- [33] J. B. Fenn. Electrospray wings for molecular elephants (nobel lecture). *Angew Chem Int Ed Engl*, 42(33):3871–3894, 2003. 9
- [34] J. Fernandez-de-Cossio Diaz and J. Fernandez-de-Cossio. Computation of isotopic peak center-mass distribution by Fourier transform. *Anal Chem*, 84(16):7052–7056, 2012. 18
- [35] A. R. Fernie, R. N. Trethewey, A. J. Krotzky and L. Willmitzer. Metabolite profiling: From diagnostics to systems biology. *Nat Rev Mol Cell Biol*, 5(9):763–769, 2004. 1, 6, 7, 12
- [36] O. Fiehn. Extending the breadth of metabolite profiling by gas chromatography coupled to mass spectrometry. *Trends Anal Chem*, 27(3):261–269, 2008. 8, 10, 17
- [37] Z. Garkani-Nejad, M. Karlovits, W. Demuth, T. Stimpfl, W. Vycudilik, M. Jalali-Heravi and K. Varmuza. Prediction of gas chromatographic retention indices of a diverse set of toxicologically relevant compounds. *J Chromatogr A*, 1028(2):287–295, 2004. 15
- [38] J. Gasteiger, W. Hanebeck and K.-P. Schulz. Prediction of mass spectra from structural information. *J Chem Inf Comput Sci*, 32(4):264–271, 1992. 19
- [39] M. Gerlich and S. Neumann. MetFusion: integration of compound identification strategies. *J Mass Spectrom*, 48(3):291–298, 2013. 20
- [40] J. Gillard, J. Frenkel, V. Devos, K. Sabbe, C. Paul, M. Rempt, D. Inzé, G. Pohnert, M. Vuylsteke, and W. Vyverman. Metabolomics enables the structure elucidation of a diatom sex pheromone. *Angew Chem Int Ed Engl*, 52(3):854–857, 2013. 3
- [41] P. Goodley. Maximizing MS/MS fragmentation in the ion trap using CID voltage ramping. Technical Report 5988-0704EN, Agilent Technologies, 2007. 12, 17
- [42] T. Griebel, M. Brinkmeyer and S. Böcker. EPoS: A modular software framework for phylogenetic analysis. *Bioinformatics*, 24(20):2399–2400, 2008. 67

- [43] J. H. Gross. *Mass Spectrometry: A textbook*. Springer, Berlin, Berlin, 2nd edition, 2011. 5, 9
- [44] J. M. Halket, D. Waterman, A. M. Przyborowska, R. K. P. Patel, P. D. Fraser and P. M. Bramley. Chemical derivatization and mass spectral libraries in metabolic profiling by GC/MS and LC/MS/MS. *J Exp Bot*, 56(410):219–243, 2005. 11, 12, 71
- [45] W. Heerma and J. J. D. Ridder. The electron-impact-induced fragmentation of some alkyl isocyanides and α -branched alkyl cyanides. *Org Mass Spectrom*, 3:1439–1456, 1970. 33, 91
- [46] M. Heinonen, A. Rantanen, T. Mielikäinen, J. Kokkonen, J. Kiuru, R. A. Ketola and J. Rousu. FiD: A software for ab initio structural identification of product ions from tandem mass spectrometric data. *Rapid Commun Mass Spectrom*, 22(19):3043–3052, 2008. 20
- [47] M. Heinonen, H. Shen, N. Zamboni and J. Rousu. Metabolite identification and molecular fingerprint prediction via machine learning. *Bioinformatics*, 28(18):2333–2341, 2012. Proc. of *European Conference on Computational Biology (ECCB 2012)*. 21
- [48] M. Herlihy, N. Shavit and M. Tzafrir. Hopscotch hashing. In *Proc. of Symposium on Distributed Computing (DISC 2008)*, volume 5218 of *Lect Notes Comput Sci*, pages 350–364. Springer, Berlin, 2008. 57
- [49] F. Hernández, T. Portolés, E. Pitarch and F. J. López. Gas chromatography coupled to high-resolution time-of-flight mass spectrometry to analyze trace-level organic compounds in the environment, food safety and toxicology. *Trends Anal Chem*, 30(2):388–400, 2011. 11, 16
- [50] M. Hesse, B. Zeeh and H. Meier. *Spectroscopic Methods in Organic Chemistry*. Thieme Medical Pub, 1997. 27, 28
- [51] A. W. Hill and R. J. Mortishire-Smith. Automated assignment of high-resolution collisionally activated dissociation mass spectra using a systematic bond disconnection approach. *Rapid Commun Mass Spectrom*, 19:3111–3118, 2005. 20
- [52] D. W. Hill, T. M. Kertesz, D. Fontaine, R. Friedman and D. F. Grant. Mass spectral metabonomics beyond elemental formula: Chemical database querying by matching experimental with computational fragmentation spectra. *Anal Chem*, 80(14):5574–5582, 2008. 60
- [53] C. Hopley, T. Bristow, A. Lubben, A. Simpson, E. Bull, K. Klagkou, J. Herniman, and J. Langley. Towards a universal product ion mass spectral library – reproducibility of product ion spectra across eleven different mass spectrometers. *Rapid Commun Mass Spectrom*, 22(12):1779–1786, 2008. 13, 17
- [54] H. Horai, M. Arita and T. Nishioka. Comparison of ESI-MS spectra in MassBank database. In *Proc. of Conference on BioMedical Engineering and Informatics (BMEI 2008)*, volume 2, pages 853–857, 2008. 15, 16

- [55] H. Horai, M. Arita, S. Kanaya, Y. Nihei, T. Ikeda, K. Suwa, Y. Ojima, K. Tanaka, S. Tanaka, K. Aoshima, Y. Oda, Y. Kakazu, M. Kusano, T. Tohge, F. Matsuda, Y. Sawada, M. Y. Hirai, H. Nakanishi, K. Ikeda, N. Akimoto, T. Maoka, H. Takahashi, T. Ara, N. Sakurai, H. Suzuki, D. Shibata, S. Neumann, T. Iida, K. Tanaka, K. Funatsu, F. Matsuura, T. Soga, R. Taguchi, K. Saito, and T. Nishioka. MassBank: A public repository for sharing mass spectral data for life sciences. *J Mass Spectrom*, 45(7):703–714, 2010. 15, 16, 60
- [56] E. C. Horning and M. G. Horning. Metabolic profiles: Gas-phase methods for analysis of metabolites. *Clin Chem*, 17(8):802–809, 1971. 2, 10
- [57] F. Hufsky and S. Böcker. Comparing fragmentation trees from electron impact mass spectra with annotated fragmentation pathways. In *Proc. of German Conference on Bioinformatics (GCB 2012)*, volume 26 of *OpenAccess Series in Informatics (OASISs)*, pages 12–22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012. ix, 33, 34
- [58] F. Hufsky, L. Kuchenbecker, K. Jahn, J. Stoye and S. Böcker. Swiftly computing center strings. In *Proc. of Workshop on Algorithms in Bioinformatics (WABI 2010)*, volume 6293 of *Lect Notes Comput Sci*, pages 325–336. Springer, Berlin, 2010. ix
- [59] F. Hufsky, L. Kuchenbecker, K. Jahn, J. Stoye and S. Böcker. Swiftly computing center strings. *BMC Bioinformatics*, 12:106, 2011. ix
- [60] F. Hufsky, K. Dührkop, F. Rasche, M. Chimani and S. Böcker. Fast alignment of fragmentation trees. *Bioinformatics*, 28:i265–i273, 2012. *Proc. of Intelligent Systems for Molecular Biology (ISMB 2012)*. ix, 47, 50
- [61] F. Hufsky, M. Rempt, F. Rasche, G. Pohnert and S. Böcker. De novo analysis of electron impact mass spectra using fragmentation trees. *Anal Chim Acta*, 739:67–76, 2012. ix, 11, 16, 25, 33, 34
- [62] F. Hufsky, K. Scheubert and S. Böcker. Computational mass spectrometry for small molecule fragmentation. *Trends Anal Chem*, 53:41–48, 2014. ix, 15
- [63] F. Hufsky, K. Scheubert and S. Böcker. New kids on the block: Novel informatics methods for natural product discovery. *Nat Prod Rep*, 2014. Accepted for publication, DOI: 10.1039/C3NP70101H. ix, 15
- [64] J. Hummel, N. Strehmel, J. Selbig, D. Walther and J. Kopka. Decision tree supported substructure prediction of metabolites from GC-MS profiles. *Metabolomics*, 6(2):322–333, 2010. 21
- [65] H. J. Issaq, Q. N. Van, T. J. Waybright, G. M. Muschik and T. D. Veenstra. Analytical and statistical approaches to metabolomics research. *J Sep Sci*, 32(13): 2183–2199, 2009. 2, 6, 15, 17, 34
- [66] N. Jaitly, M. E. Monroe, V. A. Petyuk, T. R. W. Clauss, J. N. Adkins and R. D. Smith. Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline. *Anal Chem*, 78(21):7397–7409, 2006. 26

- [67] J. Jeong, X. Shi, X. Zhang, S. Kim and C. Shen. Model-based peak alignment of metabolomic profiling from comprehensive two-dimensional gas chromatography mass spectrometry. *BMC Bioinformatics*, 13:27, 2012. 15
- [68] T. Jiang, L. Wang and K. Zhang. Alignment of trees: An alternative to tree edit. *Theor Comput Sci*, 143(1):137–148, 1995. 47, 48, 49, 50, 52
- [69] J. L. Josephs and M. Sanders. Creation and comparison of MS/MS spectral libraries using quadrupole ion trap and triple-quadrupole mass spectrometers. *Rapid Commun Mass Spectrom*, 18(7):743–759, 2004. 17
- [70] L. J. Kangas, T. O. Metz, G. Isaac, B. T. Schrom, B. Ginovska-Pangovska, L. Wang, L. Tan, R. R. Lewis, and J. H. Miller. In silico identification software (ISIS): A machine learning approach to tandem mass spectral identification of lipids. *Bioinformatics*, 28(13):1705–1713, 2012. 19, 20
- [71] P. T. Kasper, M. Rojas-Chertó, R. Mistrik, T. Reijmers, T. Hankemeier and R. J. Vreeken. Fragmentation trees for the structural characterisation of metabolites. *Rapid Commun Mass Spectrom*, 26(19):2275–2286, 2012. 23
- [72] M. Katajamaa and M. Oresic. Data processing for mass spectrometry-based metabolomics. *J Chromatogr A*, 1158(1-2):318–328, 2007. 15
- [73] A. Kerber, R. Laue, M. Meringer and K. Varmuza. MOLGEN-MS: Evaluation of low resolution electron impact mass spectra with MS classification and exhaustive structure generation. *Adv Mass Spectrom*, 15:939–940, 2001. 19
- [74] A. Kerber, R. Laue, M. Meringer and C. Rücker. Molecules in silico: The generation of structural formulae and its applications. *J Comput Chem Japan*, 3(3):85–96, 2004. 18
- [75] A. Kerber, M. Meringer and C. Rücker. CASE via MS: Ranking structure candidates by mass spectra. *Croat Chem Acta*, 79(3):449–464, 2006. 19
- [76] T. Kind and O. Fiehn. Metabolomic database annotations via query of elemental compositions: Mass accuracy is insufficient even at less than 1 ppm. *BMC Bioinformatics*, 7(1):234, 2006. 18
- [77] T. Kind and O. Fiehn. Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinformatics*, 8:105, 2007. 18
- [78] T. Kind and O. Fiehn. Advances in structure elucidation of small molecules using mass spectrometry. *Bioanal Rev*, 2(1-4):23–60, 2010. 11, 15, 17, 21, 30
- [79] T. Kind, G. Wohlgemuth, D. Y. Lee, Y. Lu, M. Palazoglu, S. Shahbaz and O. Fiehn. FiehnLib: Mass spectral and retention index libraries for metabolomics based on quadrupole and time-of-flight gas chromatography/mass spectrometry. *Anal Chem*, 81(24):10038–10048, 2009. 2, 6, 7, 11, 17, 30, 71
- [80] M. M. Koek, R. H. Jellema, J. van der Greef, A. C. Tas and T. Hankemeier. Quantitative metabolomics based on gas chromatography mass spectrometry: status and perspectives. *Metabolomics*, 7(3):307–328, 2011. 12, 71

- [81] L. Konermann, E. Ahadi, A. D. Rodriguez and S. Vahidi. Unraveling the mechanism of electrospray ionization. *Anal Chem*, 85(1):2–9, 2012. 9, 12
- [82] I. Koo, X. Zhang and S. Kim. Wavelet- and fourier-transform-based spectrum similarity approaches to compound identification in gas chromatography/mass spectrometry. *Anal Chem*, 83(14):5631–5638, 2011. 11
- [83] J. Kopka, N. Schauer, S. Krueger, C. Birkemeyer, B. Usadel, E. Bergmüller, P. Dörmann, W. Weckwerth, Y. Gibon, M. Stitt, L. Willmitzer, A. R. Fernie, and D. Steinhauser. GMD@CSB.DB: The Golm Metabolome Database. *Bioinformatics*, 21(8):1635–1638, 2005. 15, 16, 34
- [84] K.-S. Kwok, R. Venkataraghavan and F. W. McLafferty. Computer-aided interpretation of mass spectra. III. Self-training interpretive and retrieval system. *J Am Chem Soc*, 95(13):4185–4194, 1973. 21
- [85] R. L. Last, A. D. Jones and Y. Shachar-Hill. Towards the plant metabolome and beyond. *Nat Rev Mol Cell Biol*, 8:167–174, 2007. 7
- [86] S. Y. Le, R. Nussinov and J. V. Maizel. Tree graphs of RNA secondary structures and their comparisons. *Comput Biomed Res*, 22(5):461–473, 1989. 47
- [87] J. Lederberg. Topological mapping of organic molecules. *Proc Natl Acad Sci U S A*, 53(1):134–139, 1965. 19
- [88] J. Lederberg. How DENDRAL was conceived and born. In *ACM Conf. on the History of Medical Informatics, History of Medical Informatics archive*, pages 5–19, 1987. 19
- [89] S. J. Lehotay, K. Mastovska, A. Amirav, A. B. Fialkov, T. Alon, P. A. Martos, A. de Kok, and A. R. Fernández-Alba. Identification and confirmation of chemical residues in food by chromatography-mass spectrometry and other techniques. *Trends Anal Chem*, 27(11):10170–1090, 2008. 34, 71, 73
- [90] C. Lemaréchal. Lagrangian relaxation. In *Computational Combinatorial Optimization*, volume 2241 of *Lect Notes Comput Sci*, pages 112–156. Springer, Berlin, 2001. 74
- [91] J. W.-H. Li and J. C. Vederas. Drug discovery and natural products: End of an era or an endless frontier? *Science*, 325(5937):161–165, 2009. 2, 3, 7
- [92] A. Lommen and H. J. Kools. MetAlign 3.0: performance enhancement by efficient use of advances in computer hardware. *Metabolomics*, 8(4):719–726, 2012. 15
- [93] M. Ludwig, F. Hufsky, S. Elshamy and S. Böcker. Finding characteristic substructures for metabolite classes. In *Proc. of German Conference on Bioinformatics (GCB 2012)*, volume 26 of *OpenAccess Series in Informatics (OASICS)*, pages 23–38. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012. ix, 74
- [94] M. S. I. Board Members, S.-A. Sansone, T. Fan, R. Goodacre, J. L. Griffin, N. W. Hardy, R. Kaddurah-Daouk, B. S. Kristal, J. Lindon, P. Mendes, N. Morrison, B. Nikolau, D. Robertson, L. W. Sumner, C. Taylor, M. van der Werf, B. van

- Ommen, and O. Fiehn. The metabolomics standards initiative. *Nat Biotechnol*, 25(8):846–848, 2007. 16
- [95] I. Marchi, S. Rudaz and J.-L. Veuthey. Atmospheric pressure photoionization for coupling liquid-chromatography to mass spectrometry: a review. *Talanta*, 78(1): 1–18, 2009. 13
- [96] F. W. McLafferty and F. Tureček. *Interpretation of Mass Spectra*. University Science Books, Mill valley, California, fourth edition, 1993. 11, 40, 42
- [97] I. K. Mun and F. W. McLafferty. Computer methods of molecular structure elucidation from unknown mass spectra. In *Supercomputers in Chemistry*, ACS Symposium Series, chapter 9, pages 117–124. American Chemical Society, 1981. 19
- [98] S. Neumann and S. Böcker. Computational mass spectrometry for metabolomics – a review. *Anal Bioanal Chem*, 398(7):2779–2788, 2010. 11, 15, 16
- [99] V. Nevalainen and P. Vainiotalo. Electron impact induced fragmentation of dihydro-1,4-oxathiines. 1. 2,3-substituted 5,6-dihydro-1,4-oxathiines. *Org Mass Spectrom*, 21(9):543–548, 1986. 33, 37, 38, 91
- [100] W. Niessen. *Liquid Chromatography*. Marcel Dekker, second edition, 1999. 12
- [101] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and inter-laboratory transferability of a tandem mass spectral reference library: 1. Results of an Austrian multicenter study. *J Mass Spectrom*, 44(4):485–493, 2009. 13, 15, 17
- [102] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, and H. C. Köfeler. On the inter-instrument and the inter-laboratory transferability of a tandem mass spectral reference library: 2. Optimization and characterization of the search algorithm. *J Mass Spectrom*, 44(4):494–502, 2009. 15, 17
- [103] J. V. Olsen, B. Macek, O. Lange, A. Makarov, S. Horning and M. Mann. Higher-energy c-trap dissociation for peptide modification analysis. *Nat Methods*, 4(9):709–712, 2007. 12
- [104] N. R. Pace. The universal nature of biochemistry. *Proc Natl Acad Sci U S A*, 98(3): 805–808, 2001. 7
- [105] R. Pagh and F. F. Rodler. Cuckoo hashing. *J Algorithms*, 51:122–144, 2004. 57
- [106] M. Palit and G. Mallard. Fragmentation energy index for universalization of fragmentation energy in ion trap mass spectrometers for the analysis of chemical weapon convention related chemicals by atmospheric pressure ionization-tandem mass spectrometry analysis. *Anal Chem*, 81(7):2477–2485, 2009. 13
- [107] G. J. Patti, O. Yanes and G. Siuzdak. Metabolomics: The apogee of the omics trilogy. *Nat Rev Mol Cell Biol*, 13(4):263–269, 2012. 2, 12, 21

- [108] L. Pauling, A. B. Robinson, R. Teranishi and P. Cary. Quantitative analysis of urine vapor and breath by gas-liquid partition chromatography. *Proc Natl Acad Sci U S A*, 68(10):2374–2376, 1971. 2
- [109] T. Pluskal, T. Uehara and M. Yanagida. Highly accurate chemical formula prediction tool utilizing high-resolution mass spectra, MS/MS fragmentation, heuristic rules, and isotope pattern matching. *Anal Chem*, 84(10):4396–4403, 2012. 18
- [110] P. Przybylski, T. Pospieszny, A. Huczyński and B. Brzezinski. EI MS and ESI MS studies of the bisesquiterpene from cotton seeds: Gossypol and its Aza-derivatives. *J Mass Spectrom*, 43(5):680–686, 2008. 33, 91
- [111] F. Rasche. *Identification of Small Molecules using Mass Spectrometry: A fully automated pipeline proposing similar molecules and compound classes*. PhD thesis, Friedrich-Schiller-Universität Jena, Jena, Germany, 2013. ix, 4, 47, 63
- [112] F. Rasche, A. Svatoš, R. K. Maddula, C. Böttcher and S. Böcker. Computing fragmentation trees from tandem mass spectrometry data. *Anal Chem*, 83(4):1243–1251, 2011. 22, 25, 26, 27, 30, 60, 64
- [113] F. Rasche, K. Scheubert, F. Hufsky, T. Zichner, M. Kai, A. Svatoš and S. Böcker. Identifying the unknowns by aligning fragmentation trees. *Anal Chem*, 84(7):3417–3426, 2012. ix, 3, 23, 47, 50, 60, 63, 72, 73, 74
- [114] I. Rauf, F. Rasche, F. Nicolas and S. Böcker. Finding maximum colorful subtrees in practice. In *Proc. of Research in Computational Molecular Biology (RECOMB 2012)*, volume 7262 of *Lect Notes Comput Sci*, pages 213–223. Springer, Berlin, 2012. 23, 29, 30, 32, 60
- [115] M. Rempt. *Investigation of novel pathways and metabolites involved in the chemical defense of bryophytes and macroalgae*. PhD thesis, Friedrich-Schiller-Universität Jena, Jena, Germany, 2012. ix, 40
- [116] L. Ridder, J. J. J. van der Hooft, S. Verhoeven, R. C. H. de Vos, R. van Schaik and J. Vervoort. Substructure-based annotation of high-resolution multistage MSⁿ spectral trees. *Rapid Commun Mass Spectrom*, 26(20):2461–2471, 2012. 20
- [117] A. L. Rockwood and P. Haimi. Efficient calculation of accurate masses of isotopic peaks. *J Am Soc Mass Spectrom*, 17(3):415–419, 2006. 18
- [118] U. Roessner, C. Wagner, J. Kopka, R. Trethewey and L. Willmitzer. Technical advance: Simultaneous analysis of metabolites in potato tuber by gas chromatography-mass spectrometry. *Plant J*, 23:131–142, 2000. 11
- [119] M. Rojas-Chertó, P. T. Kasper, E. L. Willighagen, R. J. Vreeken, T. Hankemeier and T. H. Reijmers. Elemental composition determination based on MSⁿ. *Bioinformatics*, 27:2376–2383, 2011. 23
- [120] M. Rojas-Chertó, J. E. Peironcelly, P. T. Kasper, J. J. J. van der Hooft, R. C. H. de Vos, R. J. Vreeken, T. Hankemeier, and T. H. Reijmers. Metabolite identification using automated comparison of high-resolution multistage mass spectral trees. *Anal Chem*, 84(13):5524–5534, 2012. 23

- [121] D. H. Russell and R. D. Edmondson. High-resolution mass spectrometry and accurate mass measurements with emphasis on the characterization of peptides and proteins by matrix-assisted laser desorption/ionization time-of-flight mass spectrometry. *J Mass Spectrom*, 32:263–276, 1997. 7
- [122] D. H. Russell, M. L. Gross, J. V. D. Greef and N. M. M. Nibbering. The chemistry of C₆H₆O radical cations: A study of rearrangement reactions of halogen substituted ethyl phenyl ethers. *Org Mass Spectrom*, 14(9):474–481, 1979. 40
- [123] T. R. Sana, J. C. Roark, X. Li, K. Waddell and S. M. Fischer. Molecular formula and METLIN Personal Metabolite Database matching applied to the identification of compounds generated by LC/TOF-MS. *J Biomol Tech*, 19(4):258–266, 2008. 15
- [124] K. Scheubert, F. Hufsky, F. Rasche and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. In *Proc. of Research in Computational Molecular Biology (RECOMB 2011)*, volume 6577 of *Lect Notes Comput Sci*, pages 377–391. Springer, Berlin, 2011. ix, 30
- [125] K. Scheubert, F. Hufsky, F. Rasche and S. Böcker. Computing fragmentation trees from metabolite multiple mass spectrometry data. *J Comput Biol*, 18(11):1383–1397, 2011. ix, 22, 29
- [126] K. Scheubert, F. Hufsky and S. Böcker. Computational mass spectrometry for small molecules. *J Cheminform*, 5:12, 2013. ix, 15
- [127] B. M. Schmidt, D. M. Ribnicky, P. E. Lipsky and I. Raskin. Revisiting the ancient concept of botanical therapeutics. *Nat Chem Biol*, 3(7):360–366, 2007. 7
- [128] E. L. Schymanski and S. Neumann. The critical assessment of small molecule identification (CASMI): Challenges and solutions. *Metabolites*, 3(3):517–538, 2013. 16
- [129] E. L. Schymanski, M. Meringer and W. Brack. Matching structures to mass spectra using fragmentation patterns: Are the results as good as they look? *Anal Chem*, 81(9):3608–3617, 2009. 19
- [130] E. L. Schymanski, C. M. J. Gallampois, M. Krauss, M. Meringer, S. Neumann, T. Schulze, S. Wolf, and W. Brack. Consensus structure elucidation combining GC/EI-MS, structure generation and calculated properties. *Anal Chem*, 84(7):3287–3295, 2012. 44
- [131] D. R. Scott. Pattern recognition/expert system for mass spectra of volatile toxic and other organic compounds. *Anal Chim Acta*, 265:43–54, 1992. 21
- [132] D. R. Scott. Rapid and accurate method for estimating molecular weights of organic compounds from low resolution mass spectra. *Chemometr Intell Lab*, 16(3):193–202, 1992. 35, 72, 73, 93
- [133] D. R. Scott, A. Levitsky and S. E. Stein. Large scale evaluation of a pattern recognition/expert system for mass spectral molecular weight estimation. *Anal Chim Acta*, 278:137–147, 1993. 21

- [134] J. Senior. Partitions and their representative graphs. *Amer J Math*, 73(3):663–689, 1951. 26, 27
- [135] M. T. Sheldon, R. Mistrik and T. R. Croley. Determination of ion structures in structurally related compounds using precursor ion fingerprinting. *J Am Soc Mass Spectrom*, 20(3):370–376, 2009. 23
- [136] F. Sikora. *Aspects algorithmiques de la comparaison d’éléments biologiques*. PhD thesis, Université Paris-Est, 2011. 29
- [137] C. A. Smith, G. O’Maille, E. J. Want, C. Qin, S. A. Trauger, T. R. Brandon, D. E. Custodio, R. Abagyan, and G. Siuzdak. METLIN: A metabolite mass spectral database. *Ther Drug Monit*, 27(6):747–751, 2005. 15
- [138] C. A. Smith, E. J. Want, G. O’Maille, R. Abagyan and G. Siuzdak. XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem*, 78(3):779–787, 2006. 15
- [139] D. H. Smith, N. A. Gray, J. G. Nourse and C. W. Crandell. The DENDRAL project: Recent advances in computer-assisted structure elucidation. *Anal Chim Acta*, 133(4):471 – 497, 1981. 19
- [140] M. Sniedovich. Dijkstra’s algorithm revisited: The dynamic programming connexion. *Control Cybern*, 35(3):599–620, 2006. 53, 57
- [141] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *Univ Kans sci bull*, 38:1409–1438, 1958. 67
- [142] O. D. Sparkman. Evaluating electron ionization mass spectral library search results. *J Am Soc Mass Spectrom*, 7(4):313–318, 1996. 17
- [143] S. E. Stein. Mass spectral reference libraries: An ever-expanding resource for chemical identification. *Anal Chem*, 84(17):7274–7282, 2012. 16, 17
- [144] S. E. Stein. Chemical substructure identification by mass spectral library searching. *J Am Soc Mass Spectrom*, 6(8):644–655, 1995. 17
- [145] S. E. Stein. An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data. *J Am Soc Mass Spectrom*, 10(8):770–781, 1999. 17
- [146] S. E. Stein and D. R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *J Am Soc Mass Spectrom*, 5(9):859–866, 1994. 16
- [147] S. E. Stein, V. I. Babushok, R. L. Brown and P. J. Linstrom. Estimation of Kováts retention indices using group contributions. *J Chem Inf Model*, 47(3):975–980, 2007. 15
- [148] L. W. Sumner, A. Amberg, D. Barrett, M. Beale, R. Beger, C. Daykin, T. Fan, O. Fiehn, R. Goodacre, J. L. Griffin, T. Hankemeier, N. Hardy, J. Harnly, R. Higashi, J. Kopka, A. Lane, J. C. Lindon, P. Marriott, A. Nicholls, M. Reily, J. Thaden, and M. R. Viant. Proposed minimum reporting standards for chemical analysis. *Metabolomics*, 3(3):211–221, 2007. 16

- [149] D. L. Sweeney. Small molecules as mathematical partitions. *Anal Chem*, 75(20): 5362–5373, 2003. 20
- [150] Z. Takáts, J. M. Wiseman, B. Gologan and R. G. Cooks. Mass spectrometry sampling under ambient conditions with desorption electrospray ionization. *Science*, 306(5695): 471–473, 2004. 13
- [151] R. Tautenhahn, K. Cho, W. Uritboonthai, Z. Zhu, G. J. Patti and G. Siuzdak. An accelerated workflow for untargeted metabolomics using the METLIN database. *Nat Biotechnol*, 30(9):826–828, 2012. 15
- [152] A. Tenhosaari. Computer-assisted composition analysis of unknown compounds by simultaneous analysis of the intensity ratios of isotope patterns of the molecular ion and daughter ions in low-resolution mass spectra. *Org Mass Spectrom*, 23(4):236–39, 1988. 73
- [153] M. Thevis and W. Schänzer. Mass spectrometry in sports drug testing: Structure characterization and analytical assays. *Mass Spectrom Rev*, 26(1):79–107, 2007. 33, 91
- [154] L. Ullmann-Zeunert, A. Muck, N. Wielsch, F. Hufsky, M. A. Stanton, S. Bartram, S. Böcker, I. T. Baldwin, K. Groten, and A. Svatoš. Determination of ¹⁵N-incorporation into plant proteins and their absolute quantitation: a new tool to study nitrogen flux dynamics and protein pool sizes elicited by plant-herbivore interactions. *J Proteome Res*, 11(10):4947–4960, 2012. ix
- [155] D. Valkenburg, I. Mertens, F. Lemièrre, E. Witters and T. Burzykowski. The isotopic distribution conundrum. *Mass Spectrom Rev*, 31(1):96–109, 2012. 18
- [156] J. van der Greef and A. K. Smilde. Symbiosis of chemometrics and metabolomics: past, present, and future. *J Chemometr*, 19:376–386, 2005. 1, 2
- [157] K. Varmuza and W. Werther. Mass spectral classifiers for supporting systematic structure elucidation. *J Chem Inf Comput Sci*, 36(2):323–333, 1996. 21
- [158] R. Venkataraghavan, F. W. McLafferty and G. E. van Lear. Computer-aided interpretation of mass spectra. *Org Mass Spectrom*, 2(1):1–15, 1969. 21
- [159] C. Vidoudez and G. Pohnert. Comparative metabolomics of the diatom *Skeletonema marinoi* in different growth phases. *Metabolomics*, 8(4):654–669, 2012. 2, 32
- [160] S. G. Villas-Bôas, S. Mas, M. Akesson, J. Smedsgaard and J. Nielsen. Mass spectrometry in metabolome analysis. *Mass Spectrom Rev*, 24(5):613–646, 2005. 12
- [161] J. Watrous, P. Roach, T. Alexandrov, B. S. Heath, J. Y. Yang, R. D. Kersten, M. van der Voort, K. Pogliano, H. Gross, J. M. Raaijmakers, B. S. Moore, J. Laskin, N. Bandeira, and P. C. Dorrestein. Mass spectral molecular networking of living microbial colonies. *Proc Natl Acad Sci U S A*, 109(26):E1743–E1752, 2012. 75
- [162] W. Weckwerth. *Metabolomics — Methods and Protocols*. Humana Press, 2007. 5

-
- [163] E. Werner, J.-F. Heilier, C. Ducruix, E. Ezan, C. Junot and J.-C. Tabet. Mass spectrometry for the identification of the discriminating signals from metabolomics: Current status and future trends. *J Chromatogr B*, 871(2):143–163, 2008. 15, 19
- [164] A. Williams. Applications of computer software for the interpretation and management of mass spectrometry data in pharmaceutical science. *Curr Top Med Chem*, 2(1):99–107, 2002. 26
- [165] D. S. Wishart. Proteomics and the human metabolome project. *Expert Rev Proteomics*, 4(3):333–335, 2007. 1
- [166] S. Wolf, S. Schmidt, M. Müller-Hannemann and S. Neumann. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinformatics*, 11:148, 2010. 20, 44, 46
- [167] Z. V. I. Zaretskii, E. E. Kingston, J. H. Beynon, R. Lanber and U. P. Schlunegger. Provitamin d and vitamin d isomerizations in the mass spectrometer. translational energy release and collision-induced dissociation studies. *Org Mass Spectrom*, 20: 336–342, 1985. 44
- [168] K. Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15:205–222, 1996. 74
- [169] K. Zhang and T. Jiang. Some MAX SNP-hard results concerning unordered labeled trees. *Inform Process Lett*, 49:249–254, 1994. 48
- [170] R. Zubarev and M. Mann. On the proper use of mass accuracy in proteomics. *Mol Cell Proteomics*, 6(3):377–381, 2007. 26
- [171] R. A. Zubarev and A. Makarov. Orbitrap mass spectrometry. *Anal Chem*, 85(11): 5288–5296, 2013. 9

A Appendix

Table A.1: Compound list for the *simulated* dataset, that is, 22 reference compounds with fragmentation pathways annotated in the literature. Compound class and the respective publication reference, name of the molecule, molecular formula and molecular mass are listed in the table. Acheson *et al.* [1] describe the fragmentation of alkyl acridines. We choose two simple alkylacridines and two reduced acridines containing chlorine. From Nevalainen and Vainiotalo [99] we select four dihydro-1,4-oxathiines with fragmentation paths additionally investigated with qualitative CID measurements. From Przybylski *et al.* [110] we extract the fragmentation pathway of gossypol and from Thevis and Schänzer [153] the one of ephedrine. Further, we choose five 2,1-benzisothiazoline 2,2-dioxide nitro derivatives from Danikiewicz *et al.* [19] and seven compounds from a study on alkyl isocyanides and methyl branched alkyl cyanides by Heerma and Ridder [45].

class and reference	name	formula	molecular mass
alkyl acridines [1]	4-ethylacridine	C ₁₅ H ₁₃ N	207.1048
	4-n-propylacridine	C ₁₆ H ₁₅ N	221.1204
	quinacrine	C ₂₃ H ₃₀ ClN ₃ O	399.2077
	6,9-dichloro-2-methoxyacridine	C ₁₄ H ₉ Cl ₂ NO	277.0061
dihydro-1,4-oxathiines [99]	5,6-hydro-3-hydroxymethyl-2-methyl-1,4-oxathiine	C ₆ H ₁₀ O ₂ S	146.0402
	5,6-dihydro-2-methyl-1,4-oxathiine-3-carboxylic acid	C ₆ H ₈ O ₃ S	160.0194
	5,6-dihydro-2-methyl-1,4-oxathiine-3-carboxylic acid methylester	C ₇ H ₁₀ O ₃ S	174.0351
	5,6-dihydro-2-methyl-1,4-oxathiine-3-carboxanilide	C ₁₂ H ₁₃ NO ₂ S	235.0667
[110]	gossypol	C ₃₀ H ₃₀ O ₈	518.1941
[153]	ephedrine	C ₁₀ H ₁₅ NO	165.1154
2,1-benzisothiazoline 2,2-dioxide nitro derivatives [19]	compound6	C ₇ H ₆ N ₂ O ₄ S	214.0048
	compound9	C ₈ H ₈ N ₂ O ₄ S	228.0205
	compound10	C ₈ H ₈ N ₂ O ₄ S	228.0205
	compound18	C ₈ H ₈ N ₂ O ₄ S	228.0205
	compound14	C ₉ H ₁₀ N ₂ O ₄ S	242.0361
alkyl isocyanides & α-branched alkyl cyanides [45]	methyl isocyanide	C ₂ H ₃ N	41.0265
	ethyl isocyanide	C ₃ H ₅ N	55.0422
	isopropyl cyanide	C ₄ H ₇ N	69.0578
	n-propyl isocyanide	C ₄ H ₇ N	69.0578
	n-butyl isocyanide	C ₅ H ₉ N	83.0735
	t-butyl cyanide	C ₅ H ₉ N	83.0735
	t-butyl isocyanide	C ₅ H ₉ N	83.0735
		min	41.0265
		max	518.1941
		average	186.6339
		median	190.5699

Table A.2: Compound list for the *measured* dataset. Compound class, name of the compound, molecular formula, CAS number, molecular mass, and relative intensity of the molecular ion are listed in the table. For TMS and PFBO derivatives, we additionally give the molecular formula using our modeled element Tms and Pfb. *CAS number of the underivatized metabolite.

class	name	formula	CAS	correct mass	molecular ion (rel. intensity)	
alcohols	(Z)-pent-2-en-1-ol	C ₅ H ₁₀ O	1576-95-0	86.0732	6.31	
	phenol	C ₆ H ₆ O	108-95-2	94.0419	100	
	cis-2-methylcyclohexanol	C ₇ H ₁₄ O	7443-70-1	114.1045	2.93	
	(E)-oct-2-en-1-ol	C ₈ H ₁₆ O	18409-17-1	128.1201	1.39	
	4-nitrophenol	C ₆ H ₅ NO ₃	100-02-7	139.0269	100	
aldehydes	(E)-pent-2-enal	C ₅ H ₈ O	1576-87-0	84.0575	77.81	
	(E)-hex-2-enal	C ₆ H ₁₀ O	6728-26-3	98.0732	26.85	
	(E,E)-octa-2,4-dienal	C ₈ H ₁₂ O	5577-44-6	124.0888	13.19	
	2-ethylhexanal	C ₈ H ₁₆ O	123-05-7	128.1201	0.73	
nitrogen compounds	N,N-dimethylpyridin-4-amine	C ₇ H ₁₀ N ₂	1122-58-3	122.0844	77.88	
	1,3-dimethyl-3,4,5,6-tetrahydro-2(1H)-pyrimidinone	C ₆ H ₁₂ N ₂ O	7226-23-5	128.0950	100	
	dicyclohexylamine	C ₁₂ H ₂₃ N	101-83-7	181.1830	11.32	
	(E)-1,2-diphenyldiazene	C ₁₂ H ₁₀ N ₂	103-33-3	182.0844	32.98	
	N-phenylbenzamide	C ₁₃ H ₁₁ NO	93-98-1	197.0841	29.77	
carboxylic compounds	2-hydroxybenzoic acid	C ₇ H ₆ O ₃	69-72-7	138.0317	59.62	
	diethyl malonate	C ₇ H ₁₂ O ₄	105-53-3	160.0736	4.32	
polyaromatic compounds	naphtalene	C ₁₀ H ₈	91-20-3	128.0626	100	
	1-methylnaphthalene	C ₁₁ H ₁₀	90-12-0	142.0783	100	
	anthracene	C ₁₄ H ₁₀	120-12-7	178.0783	100	
aliphatic compounds	cyclohexene	C ₆ H ₁₀	110-83-8	82.0783	52.39	
	1,4-cyclohexadiene,1-methyl-4-(1-methylethyl)-	C ₁₀ H ₁₆	99-85-4	136.1252	21.88	
	1-methyl-4-(1-methylethenyl)cyclohexen	C ₁₀ H ₁₆	5989-27-5	136.1252	22.75	
	tetradecane	C ₁₄ H ₃₀	629-59-4	198.2348	3.55	
	octacosane	C ₂₈ H ₅₈	630-02-4	394.4539	6.32	
	nonadecane	C ₁₉ H ₄₀	629-92-5	268.3130	11.50	
halogenic compounds	parachlorophenol	C ₆ H ₅ ClO	106-48-9	128.0029	100	
	6-chlorohexan-2-one	C ₆ H ₁₁ ClO	10226-30-9	134.0498	0.89	
	(2-chloroethoxy)benzene	C ₈ H ₉ ClO	622-86-6	156.0342	31.30	
	1-chloro-4-nitrobenzene	C ₆ H ₄ ClNO ₂	100-00-5	156.9931	40.35	
	1,2-dichloro-3-nitrobenzene	C ₆ H ₃ Cl ₂ NO ₂	3209-22-1	190.9541	8.24	
fatty acid methylesters	methyl palmitate	C ₁₇ H ₃₄ O ₂	112-39-0	270.2559	21.38	
	dicranin methyl ester	C ₁₉ H ₂₈ O ₂	61481-30-9*	288.2089	0.58	
	(9Z,12Z,15Z)-methyl octadeca-9,12,15- trienoate	C ₁₉ H ₃₂ O ₂	301-00-8	292.2402	28.02	
	(9Z,12Z)-methyl octadeca-9,12-dienoate	C ₁₉ H ₃₄ O ₂	112-63-0	294.2559	39.05	
	methyl tricosanoate	C ₂₄ H ₄₈ O ₂	2433-97-8	368.3654	76.46	
steroids	ergosterol	C ₂₈ H ₄₄ O	57-87-4	396.3392	53.86	
	brassicasterol	C ₂₈ H ₄₆ O	474-67-9	398.3549	7.23	
	campesterol	C ₂₈ H ₄₈ O	474-62-4	400.3705	100.00	
	stigmasterol	C ₂₉ H ₄₈ O	83-48-7	412.3705	21.14	
aromatic, ketone	Benzophenone	C ₁₃ H ₁₀ O	119-61-9	182.0732	19.74	
TMS derivates	(E)-trimethyl(pent-2-enyloxy)silane	C ₈ H ₁₈ OSi	C ₅ H ₉ OTms	1576-96-1*	158.1127	3.03
	(E)-(hex-2-enyloxy) trimethylsilane	C ₉ H ₂₀ OSi	C ₆ H ₁₁ OTms	928-95-0*	172.1283	0.01
	(E)-trimethyl (oct-2-enyloxy)silane	C ₁₁ H ₂₄ OSi	C ₈ H ₁₅ OTms	18409-17-1*	200.1596	3.17
	(E)-trimethyl(non-2-enyloxy)silane	C ₁₂ H ₂₆ OSi	C ₉ H ₁₇ OTms	31502-14-4*	214.1753	1.88
	(9Z,12Z,15Z)-trimethylsilyl octadeca- 9,12,15-trienoate	C ₂₁ H ₃₈ O ₂ Si	C ₁₈ H ₂₉ O ₂ Tms	463-40-1*	350.2641	4.89
	trimethylsilyl nonadecanoate	C ₂₂ H ₄₆ O ₂ Si	C ₁₉ H ₃₇ O ₂ Tms	646-30-0*	370.3267	8.39
	(5Z,8Z,11Z,14Z)-trimethylsilyl icos- 5,8,11,14-tetraenoate	C ₂₃ H ₄₀ O ₂ Si	C ₂₀ H ₃₁ O ₂ Tms	506-32-1*	376.2798	1.78
PFBO derivates	(Z)-benzaldehyde O-perfluorobenzyl oxime	C ₁₄ H ₈ F ₅ NO	C ₇ H ₆ NOPfb	100-52-7*	301.0526	23.77
	(1Z,2E,4E)-octa-2,4-dienal O- perfluorobenzyl oxime	C ₁₅ H ₁₄ F ₅ NO	C ₈ H ₁₂ NOPfb	5577-44-6*	319.0996	9.39
	(1Z,2E,4E)-deca-2,4-dienal O- perfluorobenzyl oxime	C ₁₇ H ₁₈ F ₅ NO	C ₁₀ H ₁₆ NOPfb	25152-84-5*	347.1309	8.32
			min	82.0783		
			max	412.3705		
			average	212.9098		
			median	179.6306		

Table A.3: Results for the identification of the molecular ion peaks and molecular formulas for all compounds in the *measured* dataset. The rank of the correct molecular ion peak is given in the column *molecular ion*. The rank of the correct molecular formula in the list of all molecular formulas of all potential molecular ion peaks is given in the column *molecular formula*. The nominal molecular weights of the compounds estimated by Scott’s algorithm [132] are given in column *NIST MW Estimator*. We used the implementation in NIST MS Search Software version 2.0f (demo version). Challenging compounds with a molecular ion peak with relative intensity below 5 % are colored gray.

class	CAS number	molecular ion peak	exact	Fragmentation Trees		NIST MW Estimator
		(rel. intensity)	molecular mass	molecular ion	molecular formula	
alcohol	1576-95-0	6.31	86.0732	1	1	83
	108-95-2	100	94.0419	1	1	94
	7443-70-1	2.93	114.1045	1	1	114
	18409-17-1	1.39	128.1201	1	1	128
	100-02-7	100	139.0269	1	1	139
aldehyde	1576-87-0	77.81	84.0575	1	1	84
	6728-26-3	26.85	98.0732	1	1	98
	5577-44-6	13.19	124.0888	1	1	124
	123-05-7	0.73	128.1201	1	1	128
aliphatic compound	110-83-8	52.39	82.0783	2	2	94
	5989-27-5	22.75	136.1252	1	1	136
	99-85-4	21.88	136.1252	1	1	136
	629-59-4	3.55	198.2348	1	1	198
	629-92-5	11.50	268.3130	1	1	268
	630-02-4	6.32	394.4539	1	1	394
aromatic, ketone	119-61-9	19.74	182.0732	1	1	182
carboxylic compound	69-72-7	59.62	138.0317	1	1	138
	105-53-3	4.32	160.0736	1	1	160
fatty acid methylester	112-39-0	21.38	270.2559	1	1	270
	61481-30-9	0.58	288.2089	1	1	257
	301-00-8	28.02	292.2402	1	1	292
	112-63-0	39.05	294.2559	1	1	294
halogenic compound	2433-97-8	76.46	368.3654	1	1	368
	106-48-9	100	128.0029	1	1	128
	10226-30-9	0.89	134.0498	1	1	134
	622-86-6	31.30	156.0342	2	4	156
	100-00-5	40.35	156.9931	3	3	157
nitrogen compound	3209-22-1	8.24	190.9541	1	1	191
	1122-58-3	77.88	122.0844	1	1	122
	7226-23-5	100	128.0950	1	1	128
	101-83-7	11.32	181.1830	1	1	181
	103-33-3	32.98	182.0844	1	2	182
polyaromatic compound	93-98-1	29.77	197.0841	2	3	197
	91-20-3	100	128.0626	1	4	128
	90-12-0	100	142.0783	1	1	142
steroid	120-12-7	100	178.0783	1	12	178
	57-87-4	53.86	396.3392	1	4	396
	474-67-9	7.23	398.3549	1	4	398
	474-62-4	100	400.3705	2	3	400
	83-48-7	21.14	412.3705	1	1	412
PFBO derivatives	PFB-100-52-7	23.77	301.0526	1	1	301
	PFB-25152-84-5	8.32	347.1309	1	1	347
	PFB-5577-44-6	9.39	319.0996	1	1	319
TMS derivatives	TMS-1576-96-1	3.03	158.1127	1	1	158
	TMS-18409-17-1	3.17	200.1596	1	1	200
	TMS-31502-14-4	1.88	214.1753	1	1	214
	TMS-463-40-1	4.89	350.2641	1	1	350
	TMS-506-32-1	1.78	376.2798	3	4	376
	TMS-646-30-0	8.39	370.3267	1	1	370
	TMS-928-95-0	0.01	172.1283	1	1	157

Table A.4: Results of the MetFrag analysis for the 40 underivatized compounds from the *measured dataset*. Number of peaks that could be explained as fragment of the compound are given for MetFrag (*MF*) and our method (*FTs*). Number of peaks with the same explanation using both methods are given in column *both*. Results of MetFrag’s database search feature to search in PubChem (given the molecular formula of the compound) are listed. Number of *hits* in PubChem matching the molecular formula, *runtime* for fragmenting all hits and *rank* of the correct compound are given. For two compounds, no identification could be achieved, since the user session expired after two hours.

class	name	formula	mass	CAS	CID	explained fragments			MetFrag search in PubChem (molecular formula given)		
						MF	FTs	both	hits	runtime	rank
alcohols	(Z)-pent-2-en-1-ol	C ₅ H ₁₀ O	86.0732	1576-95-0	5364919	4	8	4	156	41 s	1
	phenol	C ₆ H ₆ O	94.0419	108-95-2	996	9	14	8	69	52 s	2
	cis-2-methylcyclohexanol	C ₇ H ₁₄ O	114.1045	7443-70-1	11418	10	12	9	541	8:08 min	226
	(E)-oct-2-en-1-ol	C ₈ H ₁₆ O	128.1201	18409-17-1	5318599	11	16	9	785	9 min	5
	4-nitrophenol	C ₆ H ₅ NO ₃	139.0269	100-02-7	980	8	17	6	109	2:04 min	46
aldehydes	(E)-pent-2-enal	C ₅ H ₈ O	84.0575	1576-87-0	5364752	3	8	2	171	44 s	67
	(E)-hex-2-enal	C ₆ H ₁₀ O	98.0732	6728-26-3	5281168	3	11	3	435	4:09 min	184
	(E,E)-octa-2,4-dienal	C ₈ H ₁₂ O	124.0888	5577-44-6	5283329	6	10	1	932	9:40 min	167
	2-ethylhexanal	C ₈ H ₁₆ O	128.1201	123-05-7	31241	2	8	2	785	10 min	329
	N,N-dimethylpyridin-4-amine	C ₇ H ₁₀ N ₂	122.0844	1122-58-3	14284	6	17	3	350	8:16 min	155
nitrogen compounds	1,3-dimethyl-3,4,5,6-tetrahydro-2(1H)-pyrimidinone	C ₆ H ₁₂ N ₂ O	128.0950	7226-23-5	81646	6	20	4	594	14:10 min	129
	dicyclohexylamine	C ₁₂ H ₂₃ N	181.1830	101-83-7	7582	5	15	3	470	11:35 min	80
	(E)-1,2-diphenyldiazene	C ₁₂ H ₁₀ N ₂	182.0844	103-33-3	2272	2	3	2	328	10:37 min	123
	N-phenylbenzamide	C ₁₃ H ₁₁ NO	197.0841	93-98-1	7168	5	7	4	550	10:50 min	1
	2-hydroxybenzoic acid	C ₇ H ₆ O ₃	138.0317	69-72-7	338	5	4	2	124	1:13 min	41
carboxylic compounds	diethyl malonate	C ₇ H ₁₂ O ₄	160.0736	105-53-3	7761	2	14	2	779	10:23 min	276
polyaromatic compounds	naphtalene	C ₁₀ H ₈	128.0626	91-20-3	931	2	3	2	60	51 s	36
	1-methylnaphthalene	C ₁₁ H ₁₀	142.0783	90-12-0	7002	3	12	2	94	1:37 min	35
	anthracene	C ₁₄ H ₁₀	178.0783	120-12-7	8418	4	1	0	53	48 s	15
aliphatic compounds	cyclohexene	C ₆ H ₁₀	82.0783	110-83-8	8079	3	12	3	95	41 s	81
	1,4-cyclohexadiene, 1-methyl-4-(1-methylethyl)-	C ₁₀ H ₁₆	136.1252	99-85-4	7461	6	25	5	885	12:08 min	336
	1-methyl-4-(1-methylethenyl)cyclohexen	C ₁₀ H ₁₆	136.1252	5989-27-5	22311	6	20	5	885	11:39 min	561
	tetradecane	C ₁₄ H ₃₀	198.2348	629-59-4	12389	9	17	9	118	1:23 min	4
	octacosane	C ₂₈ H ₅₈	394.4539	630-02-4	12408	22	45	16	60	5:30 min	1
	nonadecane	C ₁₉ H ₄₀	268.3130	629-92-5	12401	12	31	12	93	1:45 min	6
	parachlorophenol	C ₆ H ₅ ClO	128.0029	106-48-9	4684	5	11	5	18	22 s	6
	6-chlorohexan-2-one (2-chloroethoxy) benzene	C ₆ H ₁₁ ClO	134.0498	10226-30-9	82468	7	26	5	138	2 min	12
halogenic compounds	1-chloro-4-nitrobenzene	C ₆ H ₉ ClO	156.0342	622-86-6	12156	7	9	5	158	1:40 min	1
	1,2-dichloro-3-nitrobenzene	C ₆ H ₄ ClNO ₂	156.9931	100-00-5	7474	4	12	3	41	50 s	27
	1,2-dichloro-3-nitrobenzene	C ₆ H ₃ Cl ₂ NO ₂	190.9541	3209-22-1	18555	4	9	2	38	53 s	37
	methyl palmitate	C ₁₇ H ₃₄ O ₂	270.2559	112-39-0	8181	17	35	17	337	13:36 min	2
fatty acid methylesters	dicranin methyl ester (9Z,12Z,15Z)-methyl octadeca-9,12,15-trienoate	C ₁₉ H ₂₈ O ₂	288.2089	61481-30-9*			not included in PubChem				
	(9Z,12Z)-methyl octadeca-9,12-dienoate	C ₁₉ H ₃₂ O ₂	292.2402	301-00-8	9316	5	25	2	540	45 min	52
	(9Z,12Z)-methyl octadeca-9,12-dienoate	C ₁₉ H ₃₄ O ₂	294.2559	112-63-0	5284421	16	56	9	371	26:55 min	109
	methyl tricosanoate	C ₂₄ H ₄₈ O ₂	368.3654	2433-97-8	75519	24	32	17	121	9:36 min	1
steroids	ergosterol	C ₂₈ H ₄₄ O	396.3392	57-87-4	6433143	15	71	9	300	1:40 h	26
	brassicasterol	C ₂₈ H ₄₆ O	398.3549	474-67-9	5281327	18	117	15	357	User Session Expired	
	campesterol	C ₂₈ H ₄₈ O	400.3705	474-62-4	312822	15	85	13	272	1:21 h	15
	stigmasterol	C ₂₉ H ₄₈ O	412.3705	83-48-7	5280794	25	114	18	427	User Session Expired	
aromatic, ketone	Benzophenone	C ₁₃ H ₁₀ O	182.0732	119-61-9	3102	2	8	2	144	1:20 min	3

Table A.5: Compound list for the *Orbitrap* dataset: Compound class, compound name, PubChem ID, molecular formula, ion type, monoisotopic mass (Da), fragmentation technique, collision energies, and number of annotated losses (edges) in hypothetical fragmentation trees. Collision energies are given in electron volt for CID and arbitrary units for HCD fragmentation. If a range is given, we used a step size of 5 units within this range. Compounds with less than three (seven) annotated losses are colored red (yellow).

group	compound	PubChem ID	molecular formula	ion	monoisotopic mass	frag.method	collision energies	annotated NLS
Alkaloid	Berberine	2353	C20H18NO4+	[M+H]+	336.124	CID	35, 45	6
Alkaloid	Bicuculline	10237	C20H17NO6	[M+H]+	367.106	CID	35	25
Alkaloid	Chelidonine	10147	C20H19NO5	[M+H]+	353.126	CID	35, 45	12
Alkaloid	Cinchonine	8350	C19H22N2O	[M+H]+	294.173	CID	35, 45, 55	66
Alkaloid	Emetine	10219	C29H40N2O4	[M+H]+	480.299	CID	35, 45	62
Alkaloid	Harmane	5281404	C12H10N2	[M+H]+	182.084	CID	35, 45, 55	1
Alkaloid	Laudanosin	15548	C21H27NO4	[M+H]+	357.194	CID	35, 45, 55, 70	9
Amino acid	Alanine	602	C3H7NO2	[M-H]-	89.048	CID	5-90	0
Amino acid	Arginine	232	C6H14N4O2	[M+H]+	174.112	CID	5-80	7
Amino acid	Asparagine	236	C4H8N2O3	[M+H]+	132.053	CID	5-75	0
Amino acid	Aspartate	424	C4H7NO4	[M-H]-	133.038	CID	5-90	4
Amino acid	Cysteine	594	C3H7NO2S	[M-H]-	121.02	CID	5-90, 150	0
Amino acid	Cystine	595	C6H12N2O4S2	[M+H]+	240.024	CID	5-45	11
Amino acid	Glutamate	611	C5H9NO4	[M+H]+	147.053	CID	5-60	4
Amino acid	Glutamine	738	C5H10N2O3	[M-H]-	146.069	CID	5-90	5
Amino acid	Glycine	750	C2H5NO2	[M-H]-	75.032	HCD	5-95	0
Amino acid	Isoleucine	791	C6H13NO2	[M+H]+	131.095	CID	5-60	2
Amino acid	Leucine	857	C6H13NO2	[M+H]+	131.095	CID	5-50	2
Amino acid	Methionine	876	C5H11NO2S	[M+H]+	149.051	CID	5-55	6
Amino acid	Phenylalanine	994	C9H11NO2	[M+H]+	165.079	CID	5-45	7
Amino acid	Proline	614	C5H9NO2	[M+H]+	115.063	CID	5-90	1
Amino acid	Serine	617	C3H7NO3	[M+H]+	105.043	HCD	5-75	2
Amino acid	Threonine	205	C4H9NO3	[M-H]-	119.058	CID	5-95, 9	2
Amino acid	Tryptophan	1148	C11H12N2O2	[M-H]-	204.09	HCD	5-95	6
Amino acid	Tyrosine	1153	C9H11NO3	[M+H]+	181.074	CID	5-45	7
Amino acid	Valine	1182	C5H11NO2	[M+H]+	117.079	CID	5-90	1
Anthocyanin	CID44256802	44256802	C47H55O27+	[M+H]+	1051.293	CID	5-45	9
Anthocyanin	CID44256805	44256805	C58H65O31+	[M+H]+	1257.351	HCD	5-45	18
Anthocyanin	Delphinidin-3-rutinoside	5492231	C27H31O16+	[M+H]+	611.161	HCD	5-45	18
Benzopyran	Armentoflavone	5281600	C30H18O10	[M+H]+	538.09	CID	35, 45, 55, 70	15
Benzopyran	Bergapten	2355	C12H8O4	[M+H]+	216.042	CID	35, 45, 55, 70	10
Benzopyran	BiochaninA	5280373	C16H12O5	[M+H]+	284.068	CID	35, 45, 55, 70	19
Benzopyran	Epicatechin	72276	C15H14O6	[M+H]+	290.079	CID	35, 45, 55, 70	8
Benzopyran	Genistein	5280961	C15H10O5	[M+H]+	270.053	CID	35, 45, 55	17
Benzopyran	Kaempferol	5280863	C15H10O6	[M+H]+	286.048	CID	35, 45, 55	26
Benzopyran	Quercetin	5280343	C15H10O7	[M+H]+	302.043	CID	35, 45, 55	23
Benzopyran	Rotenone	6758	C23H22O6	[M+H]+	394.142	CID	35, 45, 55, 70	8
Benzopyran	Rutin	5280805	C27H30O16	[M+H]+	610.153	CID	35, 45, 55, 70	9
Benzopyran	Vitexinrhamnoside	5282151	C27H30O14	[M+H]+	578.164	CID	35, 45, 55, 70	13
Benzopyran	Xanthohumol	639665	C21H22O5	[M+H]+	354.147	CID	35, 45, 55, 70	3
Carboxylic acid	Anisicacid	11370	C8H8O3	[M+H]+	152.047	CID	35, 45, 55, 70	1
Carboxylic acid	Indole-3-carboxylicAcid	69867	C9H7NO2	[M+H]+	161.048	CID	35, 45, 55, 70	2
Carboxylic acid	TrimethoxycinnamicAcid	735755	C12H14O5	[M+H]+	238.084	CID	35, 45, 55, 70	16
Glucosinolate	3-Hydroxypropyl-Glucosinolate	25245521	C10H17NO10S2	[M-H]-	375.029	HCD	5-90	9
Glucosinolate	3-Methylthiopropyl-Glucosinolate	25244538	C11H19NO9S3	[M-H]-	405.022	HCD	5-90	13
Glucosinolate	4-Methoxy-3-indolylmethyl glucosinolate	656562	C17H20N2O10S2	[M-H]-	476.056	HCD	5-90	19
Glucosinolate	7-Methylthioheptyl glucosinolate	44237368	C15H27NO9S3	[M-H]-	461.085	HCD	5-90	18
Glucosinolate	8-Methylthiooctyl glucosinolate	44237373	C16H29NO9S3	[M-H]-	475.1	HCD	5, 15-55, 65-90	21
Glucosinolate	Glucosylsin	656523	C13H25NO10S3	[M-H]-	451.064	HCD	5, 15-50, 60	4
Glucosinolate	Glucoerucin	656538	C12H21NO9S3	[M-H]-	419.038	HCD	5-90	19
Glucosinolate	Glucohirsutin	44237257	C16H29NO10S3	[M-H]-	491.095	HCD	5-90	24
Glucosinolate	Glucobarbin	44237203	C15H27NO10S3	[M-H]-	477.08	HCD	5-90	28
Glucosinolate	Glucobarbin	9548621	C11H19NO10S3	[M-H]-	421.017	HCD	55-90	30
Glucosinolate	Glucomalcommiin	25244201	C17H21NO11S2	[M-H]-	479.056	HCD	5-90	25
Glucosinolate	Glucoraphanin	9548633	C12H21NO10S3	[M-H]-	435.033	HCD	5-90	8
Glucosinolate	Glucoraphenin	6443008	C12H21NO11S3	[M-H]-	451.028	HCD	5-90	16
Glucosinolate	Indolylmethyl glucosinolate	25244590	C16H18N2O9S2	[M-H]-	446.045	HCD	5-90	22
Lipid	DErySphinganine	91486	C18H39NO2	[M-H]-	301.298	CID	25	12
Lipid	DErySphingosine	5280335	C18H37NO2	[M+H]+	299.282	CID	10	1
Lipid	Phosphatidylcholine	129900	C25H54NO6P	[M+H]+	495.369	HCD	30	3
Lipid	Phosphatidylethanolamine	46891780	C39H74NO8P	[M-H]-	715.515	CID	20	6
Sugar	Cellobiose	294	C12H22O11	[M+H]+	342.116	HCD	4	10
Sugar	DP5	C30H52O26	[M+Na]+	828.275	HCD	45	16	
Sugar	DP7	C42H72O36	[M+H]+	1152.38	HCD	12	17	
Sugar	Fucose	17106	C6H12O5	[M+Na]+	164.068	CID	46	2
Sugar	Galactose	6036	C6H12O6	[M+NH4]+	180.063	HCD	12	4
Sugar	Gentiobiose	441422	C12H22O11	[M+Na]+	342.116	CID	20	6
Sugar	Lactose	6134	C12H22O11	[M+H]+	342.116	HCD	4	10
Sugar	Mannitol	6251	C6H14O6	[M+H]+	182.079	HCD	20	12
Sugar	Mannose	18950	C6H12O6	[M+H]+	180.063	CID	15	6
Sugar	Rhamnose	19233	C6H12O5	[M+Na]+	164.068	CID	46	2
Sugar	Sorbitol	5780	C6H14O6	[M+H]+	182.079	CID	20	14
Sugar	Trehalose	7427	C12H22O11	[M+Na]+	342.116	CID	20	2

Continued on next page.

Table A.5: Compound list for the *Orbitrap* dataset (continued).

group	compound	PubChem ID	molecular formula	ion	monoisotopic mass	frag.method	collision energies	annotated NLs
Zeatin	Cis-Zeatin	449093	C10H13N5O	[M+H] ⁺	219.112	CID	44	7
Zeatin	Cis-Zeatin-9-glucoside	9842892	C16H23N5O6	[M+H] ⁺	381.165	CID	17	5
Zeatin	Cis-Zeatin-o-glucoside	25244165	C16H23N5O6	[M+H] ⁺	381.165	CID	19	6
Zeatin	Cis-Zeatin-riboside	6440982	C15H21N5O5	[M+H] ⁺	351.154	CID	11	4
Zeatin	Cis-Zeatin-riboside-O-glucoside	11713250	C21H31N5O10	[M+H] ⁺	513.207	CID	20	4
Zeatin	D5-Cis-Zeatin-riboside	6440982	C15H21N5O5	[M+H] ⁺	351.154	CID	15	15
Zeatin	D5-Trans-Zeatin	449093	C10D5H8N5O	[M+H] ⁺	224.143	CID	15	8
Zeatin	D5-Trans-Zeatin-7-glucoside		C16D5H18N5O6	[M+H] ⁺	386.196	CID	14	8
Zeatin	D5-Trans-Zeatin-9-glucoside	9842892	C16D5H18N5O6	[M+H] ⁺	386.196	CID	14	10
Zeatin	D5-Trans-Zeatin-riboside	6440982	C15H21N5O5	[M+H] ⁺	351.154	CID	13	8
Zeatin	D5-Trans-Zeatin-riboside-o-glucoside	11713250	C21H31N5O10	[M+H] ⁺	513.207	CID	23	15
Zeatin	D6-isopentenyl-Adenine		C10D6H7N5	[M+H] ⁺	209.155	CID	27	4
Zeatin	D6-isopentenyl-Adenine-7-glucoside	330023	C16D6H17N5O5	[M+H] ⁺	371.208	CID	30	1
Zeatin	D6-isopentenyl-Adenine-9-glucoside	23197432	C16D6H17N5O5	[M+H] ⁺	371.208	CID	15	6
Zeatin	D6-isopentenyl-Adenosine	24405	C15D6H15N5O4	[M+H] ⁺	341.197	CID	22	4
Zeatin	Isopentenyl-Adenine		C10H13N5	[M+H] ⁺	203.117	CID	35	2
Zeatin	Isopentenyl-Adenine-7-glucoside	330023	C16H23N5O5	[M+H] ⁺	365.17	CID	14	4
Zeatin	Isopentenyl-Adenine-9-glucoside	23197432	C16H23N5O5	[M+H] ⁺	365.17	CID	14	5
Zeatin	Isopentenyl-Adenosine	24405	C15H21N5O4	[M+H] ⁺	335.159	CID	13	3
Zeatin	Trans-Zeatin	449093	C10H13N5O	[M+H] ⁺	219.112	CID	47	6
Zeatin	Trans-Zeatin-9-glucoside	9842892	C16H23N5O6	[M+H] ⁺	381.165	CID	28	5
Zeatin	Trans-Zeatin-o-glucoside	25244165	C16H23N5O6	[M+H] ⁺	381.165	CID	28	9
Zeatin	Trans-Zeatin-riboside	6440982	C15H21N5O5	[M+H] ⁺	351.154	CID	24	1
Zeatin	Trans-Zeatin-riboside-O-glucoside	11713250	C21H31N5O10	[M+H] ⁺	513.207	CID	12	5

Table A.6: Compound list for the MassBank dataset: Compound class, compound name, PubChem ID, molecular formula, monoisotopic mass (Da), collision energies (eV), and number of annotated losses (edges) in hypothetical FTs. The ion type of all compounds is $[M+H]^+$. Compounds with less than three (seven) annotated losses are colored red (yellow).

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLS
Aldehyde	1-Methoxy-3-carbaldehyde	398554	C10H9NO2	175.063	Ramp 5-60	2
Aldehyde	4-Hydroxy-3-methoxycinnamaldehyde	5280536	C10H10O3	178.063	Ramp 5-60	2
Aldehyde	Indole-3-acetaldehyde	800	C10H9NO	159.068	Ramp 5-60	2
Aldehyde	Indole-3-carboxyaldehyde	10256	C9H7NO	145.053	30, Ramp 5-60	6
Aldehyde	Syngaldehyde	8655	C9H10O4	182.058	Ramp 5-60	3
Amino acid	1-Aminocyclopropane-1-carboxylic acid	535	C4H7NO2	101.048	Ramp 5-60	0
Amino acid	2-Aminoisobutyric acid	6119	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	3-Hydroxy-DL-tryptophan	89	C10H12N2O4	224.08	Ramp 5-60	6
Amino acid	3-Methyl-L-histidine	64969	C7H11N3O2	169.085	Ramp 5-60	3
Amino acid	5-Aminovaleric acid	138	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	Alpha-Methyl-DL-histidine	4396761	C7H11N3O2	169.085	Ramp 5-60	7
Amino acid	Alpha-Methyl-DL-serine	439656	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	Carbamoyl-DL-aspartic acid	93072	C5H8N2O5	176.043	Ramp 5-60	3
Amino acid	Creatine	586	C4H9N3O2	131.069	Ramp 5-60	1
Amino acid	Cystathionine	834	C7H14N2O4S	222.067	Ramp 5-60	2
Amino acid	D-Alloisoleucine	94206	C6H13NO2	131.095	Ramp 5-60	0
Amino acid	D-beta-homophenylalanine	102530	C10H13NO2	179.095	Ramp 5-60	2
Amino acid	D-beta-homoserine	779	C4H9NO3	119.058	Ramp 5-60	4
Amino acid	Delta-Aminolevulinic acid	137	C5H9NO3	131.058	Ramp 5-60	1
Amino acid	DL-2-Aminobutyric acid	80283	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	DL-5-Hydroxylysine	1029	C6H14N2O3	162.1	Ramp 5-60	3
Amino acid	DL-alpha-epsilon-Diaminopimelic acid	865	C7H14N2O4	190.095	Ramp 5-60	4
Amino acid	DL-threo-beta-Methylaspartic acid	852	C5H9NO4	147.053	Ramp 5-60	3
Amino acid	D-Pantothenic acid	6613	C9H17NO5	219.111	Ramp 5-60	4
Amino acid	Folic acid	6037	C19H19N7O6	441.14	Ramp 5-60	4
Amino acid	Glutathione (oxidized form)	65359	C20H32N6O12S2	612.152	Ramp 5-60	13
Amino acid	Glycocyamine	763	C3H7N3O2	117.054	Ramp 5-60	1
Amino acid	Glycyl-L-proline	3013625	C7H12N2O3	172.085	Ramp 5-60	3
Amino acid	Gly-Gly	11163	C4H8N2O3	132.053	Ramp 5-60	2
Amino acid	L-(-)-Phenylalanine	6140	C9H11NO2	165.079	Ramp 5-60	4
Amino acid	L-(+)-Arginine	6322	C6H14N4O2	174.112	Ramp 5-60	1
Amino acid	L-(+)-Lysine	5962	C6H14N2O2	146.106	Ramp 5-60	0
Amino acid	L-2-Aminobutyric acid	80283	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	L-allo-threonine	99289	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	L-Asparagine	112072	C10H16N4O3	240.122	Ramp 5-60	3
Amino acid	L-Arginine	6322	C6H14N4O2	174.112	Ramp 5-60	1
Amino acid	L-beta-Homoleucine	16211048	C7H15NO2	145.11	Ramp 5-60	0
Amino acid	L-beta-homoleucine	2761525	C7H15NO2	145.11	Ramp 5-60	0
Amino acid	L-beta-homolysine	2761529	C7H16N2O2	160.121	Ramp 5-60	0
Amino acid	L-beta-homomethionine	5706673	C6H13NO2S	163.067	Ramp 5-60	2
Amino acid	L-beta-Homophenylalanine	2761537	C10H13NO2	179.095	Ramp 5-60	2
Amino acid	L-beta-homoproline	2761541	C6H11NO2	129.079	Ramp 5-60	0
Amino acid	L-beta-homoserine	1502076	C4H9NO3	119.058	Ramp 5-60	4
Amino acid	L-beta-homothreonine	5706676	C5H11NO3	133.074	Ramp 5-60	3
Amino acid	L-beta-homotryptophan	2761550	C12H14N2O2	218.106	Ramp 5-60	3
Amino acid	L-beta-homotyrosine	2761554	C10H13NO3	195.09	Ramp 5-60	2
Amino acid	L-beta-homovaline	2761558	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Carnosine	439224	C9H14N4O3	226.107	Ramp 5-60	5
Amino acid	L-Citrulline	9750	C6H13NO3	175.096	Ramp 5-60	1
Amino acid	L-Ethionine	25674	C6H13NO2S	163.067	Ramp 5-60	1
Amino acid	Leucylleucyltyrosine	88513	C21H33N3O5	407.242	Ramp 5-60	6
Amino acid	Leupeptin	439527	C20H38N6O4	426.295	Ramp 5-60	4
Amino acid	L-Glutamic acid	33032	C5H9NO4	147.053	Ramp 5-60	2
Amino acid	L-Histidine	6274	C6H9N3O2	155.069	Ramp 5-60	8
Amino acid	L-Homocarnosine	89235	C10H16N4O3	240.122	Ramp 5-60	3
Amino acid	L-Homoserine	12647	C4H9NO3	119.058	Ramp 5-60	3
Amino acid	L-Leucine	6106	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Methionine_sulfone	445282	C5H11NO4S	181.041	Ramp 5-60	2
Amino acid	L-Norleucine	21236	C6H13NO2	131.095	Ramp 5-60	1
Amino acid	L-Norvaline	65098	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	L-Proline	145742	C5H9NO2	115.063	Ramp 5-60	0
Amino acid	L-saccharopine	160556	C13H20N2O6	276.132	Ramp 5-60	4
Amino acid	L-Threonine	6288	C4H9NO3	119.058	Ramp 5-60	1
Amino acid	L-Tryptophan	6305	C11H12N2O2	204.09	Ramp 5-60	4
Amino acid	L-Tyrosine	6057	C9H11NO3	181.074	Ramp 5-60	4
Amino acid	L-Valine	6287	C5H11NO2	117.079	Ramp 5-60	0
Amino acid	N-Acetyl-DL-aspartic acid	65065	C6H9NO5	175.048	Ramp 5-60	6
Amino acid	N-Acetyl-DL-glutamic acid	70914	C7H11NO5	189.064	Ramp 5-60	6
Amino acid	N-acetyl-DL-serine	352294	C5H9NO4	147.053	Ramp 5-60	2
Amino acid	N-Acetylglycine	10972	C4H7NO3	117.043	Ramp 5-60	2
Amino acid	N-alpha-Acetyl-L-ornithine	439232	C7H14N2O3	174.1	Ramp 5-60	2
Amino acid	N-Formyl-L-methionine	439750	C6H11NO3S	177.046	Ramp 5-60	3
Amino acid	N-N-Dimethylglycine	673	C4H9NO2	103.063	Ramp 5-60	0
Amino acid	N-Tigloylglycine	6441567	C7H11NO3	157.074	Ramp 5-60	4
Amino acid	O-Phospho-L-serine	68841	C3H8NO6P	185.009	Ramp 5-60	2
Amino acid	S-Adenosyl-L-homocysteine	439155	C14H20N6O5S	384.122	Ramp 5-60	1
Amino acid	S-Lactoylglutathione	440018	C13H21N3O8S	379.105	Ramp 5-60	18
Amino acid	S-Sulfofocysteine	115015	C3H7NO5S2	200.977	Ramp 5-60	6
Benzimidazole	Thiabenzazole	5430	C10H7N3S	201.036	Ramp 5-60	3
Bile acid	Cholate	221493	C24H40O5	408.288	30, Ramp 5-60	10
Bile acid	Deoxycholate	440355	C24H40O4	392.293	30, Ramp 5-60	6
Capsaicinoid	Capsaicin	1548943	C18H27NO3	305.199	Ramp 5-60	1
Capsaicinoid	Dihydrocapsaicin	107982	C18H29NO3	307.215	Ramp 5-60	1

Continued on next page.

Table A.6: Compound list for the *MassBank* dataset (continued).

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLS
Carboxylic acid	(-)-Citramalic_acid	439766	C5H8O5	148.037	Ramp 5-60	4
Carboxylic acid	(-)-Shikimic_acid	8742	C7H10O5	174.053	Ramp 5-60	7
Carboxylic acid	(+)-Alpha-Lipoic_acid	864	C8H14O2S2	206.044	Ramp 5-60	5
Carboxylic acid	(R)-(-)-mandelic_acid	11914	C8H8O3	152.047	Ramp 5-60	1
Carboxylic acid	(S)-(+)-Citramalic_acid	441696	C5H8O5	148.037	Ramp 5-60	3
Carboxylic acid	16-Hydroxyhexadecanoic_acid	10466	C16H32O3	272.235	Ramp 5-60	0
Carboxylic acid	1-O-b-D-glucopyranosyl_sinapate	5280406	C17H22O10	386.121	Ramp 5-60	10
Carboxylic acid	2-5-Dihydroxy_benzoic_acid	3469	C7H6O4	154.027	Ramp 5-60	2
Carboxylic acid	2-Aminoethylphosphonic_acid	339	C2H8NO3P	125.024	Ramp 5-60	2
Carboxylic acid	2-Hydroxyisobutyric_acid	11671	C4H8O3	104.047	30, Ramp 5-60	2
Carboxylic acid	2-Hydroxyisocaproic_acid	439960	C6H12O3	132.079	Ramp 5-60	2
Carboxylic acid	2-Isopropylmalic_acid	5280523	C7H12O5	176.068	Ramp 5-60	5
Carboxylic acid	2-Methylglutaric_Acid	12046	C6H10O4	146.058	Ramp 5-60	2
Carboxylic acid	2-Oxobutyrate	58	C4H6O3	102.032	Ramp 5-60	0
Carboxylic acid	2-Oxovaleric_acid	74563	C5H8O3	116.047	Ramp 5-60	0
Carboxylic acid	3-4-Dihydroxybenzoic_acid	72	C7H6O4	154.027	Ramp 5-60	2
Carboxylic acid	3-Guanidinopropionic_acid	67701	C4H9N3O2	131.069	Ramp 5-60	1
Carboxylic acid	3-Hydroxy-3-methylglutarate	1662	C6H10O5	162.053	Ramp 5-60	3
Carboxylic acid	3-Hydroxymandelic_acid	86957	C8H8O4	168.042	Ramp 5-60	2
Carboxylic acid	3-Indoleacetic_acid	802	C10H9NO2	175.063	Ramp 5-60	2
Carboxylic acid	4-Coumaric_acid	637542	C9H8O3	164.047	30, Ramp 5-60	2
Carboxylic acid	4-Hydroxy-3-methoxycinnamic_acid	445858	C10H10O4	194.058	Ramp 5-60	3
Carboxylic acid	4-Hydroxy-benzoate	135	C7H6O3	138.032	Ramp 5-60	1
Carboxylic acid	6-Hydroxynicotinic_Acid	72924	C6H5NO3	139.027	Ramp 5-60	1
Carboxylic acid	Anthranilic_acid	227	C7H7NO2	137.048	Ramp 5-60	1
Carboxylic acid	Caffeic_acid	689043	C9H8O4	180.042	Ramp 5-60	2
Carboxylic acid	Cis-Aconitic_Acid	643757	C6H6O6	174.016	Ramp 5-60	3
Carboxylic acid	Citraconic_Acid	643798	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Citric_acid	311	C6H8O7	192.027	Ramp 5-60	6
Carboxylic acid	D-(-)-Quinic_acid	6508	C7H12O6	192.063	Ramp 5-60	1
Carboxylic acid	D-(+)-Galacturonic_acid	439215	C6H10O7	194.043	Ramp 5-60	11
Carboxylic acid	D-(+)-Glyceric_acid	439194	C3H6O4	106.027	Ramp 5-60	2
Carboxylic acid	D-(+)-Malic_acid	92824	C4H6O5	134.022	Ramp 5-60	4
Carboxylic acid	D-Gluconic_acid	10690	C6H12O7	196.058	Ramp 5-60	8
Carboxylic acid	D-Glucuronic_acid	94715	C6H10O7	194.043	Ramp 5-60	10
Carboxylic acid	DL-2-Hydroxyvaleric_acid	98009	C5H10O3	118.063	Ramp 5-60	1
Carboxylic acid	DL-3-4-Dihydroxymandelic_acid	85782	C8H8O5	184.037	Ramp 5-60	2
Carboxylic acid	DL-3-Aminoisobutyric_acid	64956	C4H9NO2	103.063	Ramp 5-60	1
Carboxylic acid	DL-4-Hydroxy-3-methoxymandelic_acid	1245	C9H10O5	198.053	Ramp 5-60	1
Carboxylic acid	DL-beta-Aminobutyric_acid	2761506	C4H9NO2	103.063	Ramp 5-60	0
Carboxylic acid	DL-beta-Hydroxybutyric_acid	441	C4H8O3	104.047	Ramp 5-60	1
Carboxylic acid	DL-Glyceric_acid	439194	C3H6O4	106.027	Ramp 5-60	2
Carboxylic acid	DL-Lactic_acid	107689	C3H6O3	90.032	Ramp 5-60	0
Carboxylic acid	DL-mandelic_acid	1292	C8H8O3	152.047	Ramp 5-60	1
Carboxylic acid	DL-p-Hydroxyphenyllactic_acid	9378	C9H10O4	182.058	Ramp 5-60	5
Carboxylic acid	DL-Pipecolinic_acid	439227	C6H11NO2	129.079	Ramp 5-60	0
Carboxylic acid	D-tartaric_acid	439655	C4H6O6	150.016	Ramp 5-60	4
Carboxylic acid	Gamma-Ulenolic_acid	5280933	C18H30O2	278.225	Ramp 5-60	1
Carboxylic acid	Gibberellin_A4	443457	C19H24O5	332.162	Ramp 5-60	8
Carboxylic acid	Glutaric_acid	743	C5H8O4	132.042	Ramp 5-60	2
Carboxylic acid	Homogentisic_acid	780	C8H8O4	168.042	Ramp 5-60	3
Carboxylic acid	Indole-3-carboxylic_acid	69867	C9H7NO2	161.048	Ramp 5-60	1
Carboxylic acid	Isoquavacine	3765	C6H9NO2	127.063	Ramp 5-60	1
Carboxylic acid	Isonicotinic_acid	5922	C6H5NO2	123.032	Ramp 5-60	1
Carboxylic acid	Itaconic_acid	811	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Kynurenic_acid	3845	C10H7NO3	189.043	Ramp 5-60	1
Carboxylic acid	L-(+)-Tartaric_acid	444305	C4H6O6	150.016	Ramp 5-60	2
Carboxylic acid	L-2-Aminoadipic_Acid	92136	C6H11NO4	161.069	Ramp 5-60	3
Carboxylic acid	L-Pyrogutamic_acid	7405	C5H7NO3	129.043	Ramp 5-60	0
Carboxylic acid	Maleic_acid	444266	C4H4O4	116.011	Ramp 5-60	1
Carboxylic acid	Mesaconic_acid	638129	C5H6O4	130.027	Ramp 5-60	1
Carboxylic acid	Methylsuccinic_acid	10349	C5H8O4	132.042	30, Ramp 5-60	1
Carboxylic acid	Mucic_acid	3037582	C6H10O8	210.038	Ramp 5-60	5
Carboxylic acid	N-acetylneuraminic_acid	439197	C11H19NO9	309.106	Ramp 5-60	3
Carboxylic acid	Nicotinic_Acid	938	C6H5NO2	123.032	Ramp 5-60	1
Carboxylic acid	Orotic_acid	967	C5H4N2O4	156.017	Ramp 5-60	1
Carboxylic acid	Phosphoenolpyruvic_Acid	1005	C3H5O6P	167.982	Ramp 5-60	1
Carboxylic acid	Prostaglandin_E1	5280723	C20H34O5	354.241	Ramp 5-60	6
Carboxylic acid	Rosmarinic_acid	639655	C18H16O8	360.085	Ramp 5-60	8
Carboxylic acid	Sebacic_acid	5192	C10H18O4	202.121	Ramp 5-60	3
Carboxylic acid	Sinapic_acid	637775	C11H12O5	224.068	Ramp 5-60	10
Carboxylic acid	Sinapoyl_malate	11953815	C15H16O9	340.079	Ramp 5-60	12
Carboxylic acid	Succinic_acid	1110	C4H6O4	118.027	Ramp 5-60	2
Carboxylic acid	Trans-4-Hydroxy-L-proline	5810	C5H9NO3	131.058	Ramp 5-60	2
Carboxylic acid	Trans-Cinnamic_acid	444539	C9H8O2	148.052	Ramp 5-60	1
Carboxylic acid	Urocanic_acid	736715	C6H6N2O2	138.043	Ramp 5-60	1
Coumarin	4-Methylumbelliferone	5280567	C10H8O3	176.047	Ramp 5-60	5
Coumarin	6-7-Dihydroxycoumarin	5281416	C9H6O4	178.027	30, Ramp 5-60	19
Coumarin	7-Hydroxy-4-methylcoumarin	5280567	C10H8O3	176.047	30, Ramp 5-60	10
Coumarin	Daphnetin	5280569	C9H6O4	178.027	30, Ramp 5-60	12
Coumarin	Esculin	5281417	C15H16O9	340.079	Ramp 5-60	4
Coumarin	Scopoletin	5280460	C10H8O4	192.042	Ramp 5-60	4
Ethanolamine	O-Phosphorylethanolamine	1015	C2H8NO4P	141.019	Ramp 5-60	1

Continued on next page.

Table A.6: Compound list for the *MassBank* dataset (continued).

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLS
Flavonoid	(-)-Epicatechin	72276	C15H14O6	290.079	Ramp 5-60	25
Flavonoid	(-)-Riboflavin	493570	C17H20N4O6	376.138	Ramp 5-60	4
Flavonoid	(+)-Catechin	9064	C15H14O6	290.079	Ramp 5-60	13
Flavonoid	(+)-Epicatechin	182232	C15H14O6	290.079	Ramp 5-60	13
Flavonoid	7-Methylquercetin-3-Galactoside-6-Rhamnoside-3-Rhamnoside	44259338	C34H42O20	770.227	30, Ramp 5-60	4
Flavonoid	Apigenin	5280443	C15H10O5	270.053	Ramp 5-60	2
Flavonoid	Apigenin-7-O-glucoside	5280704	C21H20O10	432.106	Ramp 5-60	7
Flavonoid	Baicalin	64982	C21H18O11	446.085	Ramp 5-60	3
Flavonoid	Daidzein	5281708	C15H10O4	254.058	30, Ramp 5-60	18
Flavonoid	Daidzin	107971	C21H20O9	416.111	Ramp 5-60	10
Flavonoid	Datiscin	5883291	C27H30O15	594.158	30, Ramp 5-60	14
Flavonoid	Eriodictyol	440735	C15H12O6	288.063	Ramp 5-60	5
Flavonoid	Eriodictyol-7-O-glucoside	5319853	C21H22O11	450.116	Ramp 5-60	7
Flavonoid	Flavanomarein	101781	C21H22O11	450.116	Ramp 5-60	4
Flavonoid	Formononetin	5280378	C16H12O4	268.074	Ramp 5-60	7
Flavonoid	Fortunellin	5317385	C28H32O14	592.179	Ramp 5-60	2
Flavonoid	Gossypin	5281621	C21H20O13	480.09	Ramp 5-60	7
Flavonoid	Hesperidin	10621	C28H34O15	610.19	Ramp 5-60	5
Flavonoid	Homoorientin	114776	C21H20O11	448.101	Ramp 5-60	13
Flavonoid	Hyperoside	5281643	C21H20O12	464.095	Ramp 5-60	8
Flavonoid	Isorhamnetin	5281654	C16H12O7	316.058	Ramp 5-60	3
Flavonoid	Isorhamnetin-3-Galactoside-6-Rhamnoside	44259338	C28H32O16	624.169	30, Ramp 5-60	8
Flavonoid	Isorhamnetin-3-O-glucoside	5318645	C22H22O12	478.111	30, Ramp 5-60	13
Flavonoid	Isorhamnetin-3-O-rutinoside	5481663	C28H32O16	624.169	30, Ramp 5-60	8
Flavonoid	Kaempferide	5281666	C16H12O6	300.063	Ramp 5-60	10
Flavonoid	Kaempferol	5280863	C15H10O6	286.048	Ramp 5-60	3
Flavonoid	Kaempferol-3-7-O-bis-alpha-L-rhamnoside	5323562	C27H30O14	578.164	30, Ramp 5-60	10
Flavonoid	Kaempferol-3-Galactoside-6-Rhamnoside-3-Rhamnoside	5281693	C33H40O19	740.216	30, Ramp 5-60	4
Flavonoid	Kaempferol-3-O-glucoside-2-p-coumaroyl	25245527	C30H26O13	594.137	Ramp 5-60	6
Flavonoid	Kaempferol-3-Glucoside-2-Rhamnoside-7-Rhamnoside	25202803	C33H40O19	740.216	30, Ramp 5-60	7
Flavonoid	Kaempferol-3-Rhamnoside-3-Rhamnoside	25202803	C27H30O15	594.158	Ramp 5-60	4
Flavonoid	Kaempferol-3-Glucoside-6-p-coumaroyl	5320686	C30H26O13	594.137	30, Ramp 5-60	11
Flavonoid	Kaempferol-3-O-glucuronide	5318759	C21H18O12	462.08	Ramp 5-60	3
Flavonoid	Kaempferol-3-O-alpha-L-arabinoside	5481882	C20H18O10	418.09	Ramp 5-60	7
Flavonoid	Kaempferol-3-O-alpha-L-rhamnopyranosyl[(1-2)-beta-D-glucopyranoside-7-O-alpha-L-rhamnopyranoside]	44258837	C33H40O19	740.216	30, Ramp 5-60	8
Flavonoid	Kaempferol-3-O-alpha-L-rhamnoside	5316673	C21H20O10	432.106	Ramp 5-60	9
Flavonoid	Kaempferol-3-O-beta-D-galactoside-7-O-alpha-L-rhamnoside	5281693	C27H30O15	594.158	30, Ramp 5-60	13
Flavonoid	Kaempferol-3-O-beta-glucopyranosyl-7-O-alpha-L-rhamnopyranoside	25203808	C27H30O15	594.158	30, Ramp 5-60	11
Flavonoid	Kaempferol-3-O-glucoside	5282102	C21H20O11	448.101	30, Ramp 5-60	13
Flavonoid	Kaempferol-3-O-rutinoside	5318767	C27H30O15	594.158	30, Ramp 5-60	6
Flavonoid	Kaempferol-3-Rhamnoside-4-Rhamnoside-7-Rhamnoside	44259005	C33H40O18	724.221	Ramp 5-60	6
Flavonoid	Kaempferol-7-O-alpha-L-rhamnoside	5316673	C21H20O10	432.106	30, Ramp 5-60	28
Flavonoid	Kaempferol-7-O-neohesperidoside	5483905	C27H30O15	594.158	30, Ramp 5-60	3
Flavonoid	Linarin	5317025	C28H32O14	592.179	Ramp 5-60	2
Flavonoid	Luteolin	5280445	C15H10O6	286.048	30, Ramp 5-60	19
Flavonoid	Luteolin-3-7-di-O-glucoside	5490298	C27H30O16	610.153	Ramp 5-60	3
Flavonoid	Luteolin-4-O-glucoside	5319116	C21H20O11	448.101	Ramp 5-60	6
Flavonoid	Luteolin-7-O-glucoside	5280637	C21H20O11	448.101	Ramp 5-60	8
Flavonoid	Marein	6441269	C21H22O11	450.116	Ramp 5-60	8
Flavonoid	Marittimein	6450184	C21H20O11	448.101	Ramp 5-60	3
Flavonoid	Myricetin-3-Galactoside	5491408	C21H20O13	480.09	Ramp 5-60	11
Flavonoid	Myricetin-3-Rhamnoside	5281673	C21H20O12	464.095	Ramp 5-60	12
Flavonoid	Myricetin-3-Xyloside	5281673	C20H18O12	450.08	Ramp 5-60	9
Flavonoid	Myricitrin	5281673	C21H20O12	464.095	Ramp 5-60	11
Flavonoid	Naringenin-7-O-glucoside	92794	C21H22O10	434.121	Ramp 5-60	7
Flavonoid	Neodiosmin	44258230	C28H32O15	608.174	Ramp 5-60	2
Flavonoid	Ononin	442813	C22H22O9	430.126	30, Ramp 5-60	5
Flavonoid	Peltatoside	5484066	C26H28O16	596.138	30, Ramp 5-60	18
Flavonoid	Poncirin	442456	C28H34O14	594.195	30, Ramp 5-60	5
Flavonoid	Procyanidin_B1	11250133	C30H26O12	578.142	Ramp 5-60	15
Flavonoid	Procyanidin_B2	122738	C30H26O12	578.142	Ramp 5-60	16
Flavonoid	Puerarin	5281807	C21H20O9	416.111	Ramp 5-60	6
Flavonoid	Quercetin	5280343	C15H10O7	302.043	Ramp 5-60	8
Flavonoid	Quercetin-3-[6-malonyl]-Glucoside	5282159	C24H22O15	550.096	Ramp 5-60	8
Flavonoid	Quercetin-3-4-O-di-beta-glucopyranoside	5320835	C27H30O17	626.148	30, Ramp 5-60	9
Flavonoid	Quercetin-3-7-O-alpha-L-dirhamnopyranoside	44259217	C27H30O15	594.158	30, Ramp 5-60	10
Flavonoid	Quercetin-3-Arabinoside	5481224	C20H18O11	434.085	Ramp 5-60	8
Flavonoid	Quercetin-3-b-xyloside	5320863	C20H18O11	434.085	Ramp 5-60	9
Flavonoid	Quercetin-3-Glucuronide	5274585	C21H18O13	478.075	Ramp 5-60	8
Flavonoid	Quercetin-3-O-alpha-L-rhamnopyranoside	5280459	C21H20O11	448.101	Ramp 5-60	12
Flavonoid	Quercetin-3-O-alpha-L-rhamnopyranosyl[(1-2)-beta-D-glucopyranoside-7-O-alpha-L-rhamnopyranoside]	5489459	C33H40O20	756.211	30, Ramp 5-60	8
Flavonoid	Quercetin-3-O-beta-D-galactoside	5281643	C21H20O12	464.095	Ramp 5-60	9
Flavonoid	Quercetin-3-O-beta-glucopyranoside	5280804	C21H20O12	464.095	Ramp 5-60	6
Flavonoid	Quercetin-3-O-beta-glucopyranosyl-7-O-alpha-L-rhamnopyranoside	5280805	C27H30O16	610.153	30, Ramp 5-60	11
Flavonoid	Quercetin-3-O-glucose-6-acetate	5280804	C23H22O13	506.106	Ramp 5-60	8
Flavonoid	Quercetin-7-O-rhamnoside	5748601	C16H12O7	448.101	Ramp 5-60	9
Flavonoid	Rhamnetin	5281691	C16H12O7	316.058	Ramp 5-60	6
Flavonoid	Rhoifolin	5282150	C27H30O14	578.164	30, Ramp 5-60	3
Flavonoid	Robinin	5281693	C33H40O19	740.216	30, Ramp 5-60	7
Flavonoid	Spiraeoside	5320844	C21H20O12	464.095	Ramp 5-60	6
Flavonoid	Syringetin-3-O-galactoside	5321576	C23H24O13	508.122	30, Ramp 5-60	17
Flavonoid	Syringetin-3-O-glucoside	5321577	C23H24O13	508.122	Ramp 5-60	14
Flavonoid	Tilliroside	5320686	C30H26O13	594.137	30, Ramp 5-60	9
Flavonoid	Vitexin	5280441	C21H20O10	432.106	Ramp 5-60	4
Flavonoid	Vitexin-2-O-rhamnoside	5282151	C27H30O14	578.164	Ramp 5-60	5
Glucosinolate	4-Methylsulfinylbutyl glucosinolate	9548634	C12H23NO10S3	437.048	Ramp 5-60	6
Glucosinolate	4-Methylthiobutyl glucosinolate	9548895	C12H23NO9S3	421.053	Ramp 5-60	4
Glucosinolate	Sinigrin	6911854	C10H17NO9S2	359.034	Ramp 5-60	4

Continued on next page.

Table A.6: Compound list for the *MassBank* dataset (continued).

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLS
Indole	3-Indoxylsulfate	10258	C8H7NO4S	213.01	Ramp 5-60	3
Indole	Harmaline	5280951	C13H14N2O	214.111	Ramp 5-60	4
Isoprenoid	Glycyrrhizic_acid	14982	C42H62O16	822.404	30, Ramp 5-60	3
Isoprenoid	Glycyrrhizin	14982	C42H62O16	822.404	30, Ramp 5-60	3
Nucleotide	1-3-Dimethylurate	70346	C7H8N4O3	196.06	30, Ramp 5-60	10
Nucleotide	1-7-Dimethylxanthine	4687	C7H8N4O2	180.065	Ramp 5-60	5
Nucleotide	2-Deoxyadenosine-5-monophosphate	12599	C10H14N5O6P	331.068	Ramp 5-60	5
Nucleotide	2-Deoxycytidine	13711	C9H13N3O4	227.091	Ramp 5-60	3
Nucleotide	2-Deoxycytidine-5-diphosphate	150855	C9H15N3O10P2	387.023	Ramp 5-60	6
Nucleotide	2-Deoxyguanosine_5-monophosphate	65059	C10H14N5O7P	347.063	Ramp 5-60	4
Nucleotide	2-Deoxyguanosine-5-diphosphate	439220	C10H15N5O10P2	427.029	Ramp 5-60	2
Nucleotide	2-Deoxyinosine-5-monophosphate	91531	C10H13N4O7P	332.052	Ramp 5-60	6
Nucleotide	2-Deoxyuridine-5-monophosphate	65063	C9H13N2O8P	308.041	Ramp 5-60	5
Nucleotide	3-Hydroxypyridine	7971	C5H5NO	95.037	30, Ramp 5-60	1
Nucleotide	3-Methylxanthine	70639	C6H6N4O2	166.049	Ramp 5-60	5
Nucleotide	4-Pyridoxate	6723	C8H9NO4	183.053	Ramp 5-60	2
Nucleotide	5-Aminoimidazole-4-carboxamide-1-beta-D-ribofuranosyl_5-monophosphate	65110	C9H15N4O8P	338.063	Ramp 5-60	3
Nucleotide	5-Deoxy-5-Methylthioadenosine	439176	C11H15N5O3S	297.09	Ramp 5-60	2
Nucleotide	6-(Gamma-gamma-Dimethylallylamino)purine	92180	C10H13N5	203.117	Ramp 5-60	6
Nucleotide	6-(Gamma-gamma-Dimethylallylamino)purine_ribose	24405	C15H21N5O4	335.159	Ramp 5-60	4
Nucleotide	Adenine	190	C5H5N5	135.054	30, Ramp 5-60	4
Nucleotide	Adenosine	60961	C10H13N5O4	267.097	Ramp 5-60	2
Nucleotide	Adenosine_3-monophosphate	41211	C10H14N5O7P	347.063	Ramp 5-60	4
Nucleotide	Adenosine_5-diphosphate	6022	C10H15N5O10P2	427.029	Ramp 5-60	3
Nucleotide	Adenosine_5-diphospho-glucose	16500	C16H25N5O15P2	589.082	Ramp 5-60	9
Nucleotide	Adenosine_5-monophosphate	6083	C10H14N5O7P	347.063	Ramp 5-60	3
Nucleotide	Beta-Nicotinamide_adenine_dinucleotide	5893	C21H27N7O14P2	663.109	Ramp 5-60	10
Nucleotide	Cytidine	6175	C9H13N3O5	243.086	Ramp 5-60	4
Nucleotide	Cytidine_5-diphosphocholine	13804	C14H26N4O11P2	488.107	Ramp 5-60	8
Nucleotide	Cytidine-3-5-cyclicmonophosphate	19236	C9H12N3O7P	305.041	Ramp 5-60	6
Nucleotide	Cytidine-3-monophosphate	66535	C9H14N3O8P	323.052	Ramp 5-60	4
Nucleotide	Cytidine-5-diphosphate	6132	C9H15N3O11P2	403.018	Ramp 5-60	5
Nucleotide	Cytidine-5-monophosphate	6131	C9H14N3O8P	323.052	Ramp 5-60	3
Nucleotide	Guanine	764	C5H5N5O	151.049	Ramp 5-60	3
Nucleotide	Guanosine	6802	C10H13N5O5	283.092	Ramp 5-60	3
Nucleotide	Guanosine_5-diphosphate-D-mannose	18396	C16H25N5O16P2	605.077	Ramp 5-60	6
Nucleotide	Guanosine_5-diphospho-beta-L-fucose	10918995	C16H25N5O15P2	589.082	Ramp 5-60	9
Nucleotide	Guanosine_5-diphosphoglucose	439225	C16H25N5O16P2	605.077	Ramp 5-60	7
Nucleotide	Guanosine_5-monophosphate	6804	C10H14N5O8P	363.058	Ramp 5-60	5
Nucleotide	Guanosine-3-5-cyclic_monophosphate	24316	C10H12N5O7P	345.047	Ramp 5-60	6
Nucleotide	Inosine	6021	C10H12N4O5	268.081	Ramp 5-60	3
Nucleotide	Inosine-5-diphosphate	6831	C10H14N4O11P2	428.013	Ramp 5-60	7
Nucleotide	Inosine-5-monophosphate	8582	C10H13N4O8P	348.047	Ramp 5-60	6
Nucleotide	N-6-(delta-2-isopentenyl)adenosine	24405	C15H21N5O4	335.159	Ramp 5-60	5
Nucleotide	Oxypurinol	4644	C5H4N4O2	152.033	Ramp 5-60	1
Nucleotide	Pyridoxal	1050	C8H9NO3	167.058	Ramp 5-60	3
Nucleotide	Pyridoxal_5-phosphate	1051	C8H10NO6P	247.025	Ramp 5-60	2
Nucleotide	Pyridoxamine	1052	C8H12N2O2	168.09	Ramp 5-60	9
Nucleotide	Pyridoxine	1054	C8H11NO3	169.074	Ramp 5-60	7
Nucleotide	Thiamine	1130	C12H17N4O5	265.112	Ramp 5-60	5
Nucleotide	Thymidine-5-diphosphate	164628	C10H16N2O11P2	402.023	Ramp 5-60	8
Nucleotide	Thymidine-5-monophosphate	9700	C10H15N2O8P	322.057	Ramp 5-60	5
Nucleotide	Thymine	1135	C5H6N2O2	126.043	Ramp 5-60	0
Nucleotide	Trans-Zeatin	449093	C10H13N5O	219.112	Ramp 5-60	8
Nucleotide	Trans-Zeatin-riboside	6440982	C15H21N5O5	351.154	Ramp 5-60	6
Nucleotide	UDP-beta-L-rhamnose	23724469	C15H24N2O16P2	550.06	Ramp 5-60	13
Nucleotide	UDP-Galactose	23724458	C15H24N2O17P2	566.055	Ramp 5-60	13
Nucleotide	UDP-xylose	23724459	C14H22N2O16P2	536.044	Ramp 5-60	15
Nucleotide	Uracil	1174	C4H4N2O2	112.027	Ramp 5-60	0
Nucleotide	Uridine	6029	C9H12N2O6	244.07	Ramp 5-60	5
Nucleotide	Uridine_5-diphosphate	6031	C9H14N2O11P2	404.002	Ramp 5-60	5
Nucleotide	Uridine_5-diphospho-D-glucose	8629	C15H24N2O17P2	566.055	Ramp 5-60	13
Nucleotide	Uridine_5-diphosphoglucuronic_acid	17473	C15H22N2O18P2	580.034	Ramp 5-60	14
Nucleotide	Uridine_5-diphospho-N-acetylglucosamine	23724461	C17H27N3O17P2	607.082	30, Ramp 5-60	17
Nucleotide	Uridine_5-diphospho-N-acetylglucosamine	445675	C17H27N3O17P2	607.082	30, Ramp 5-60	17
Nucleotide	Uridine_5-monophosphate	6030	C9H13N2O9P	324.036	Ramp 5-60	4
Nucleotide	Xanthine	1188	C5H4N4O2	152.033	Ramp 5-60	1
Nucleotide	Xanthosine	64959	C10H12N4O6	284.076	Ramp 5-60	2
Nucleotide	Xanthosine-5-monophosphate	73323	C10H13N4O9P	364.042	Ramp 5-60	6
Organosulfonic acid	2-Mercaptoethanesulfonic_acid	598	C2H6O3S2	141.976	Ramp 5-60	2
Organosulfonic acid	Hypotaurine	107812	C2H7NO2S	109.02	Ramp 5-60	2
Organosulfonic acid	5-Sulforaphane	6433206	C6H9NO2S	175.013	Ramp 5-60	4
Penicillin	Piperacillin	6604563	C23H27N5O7S	517.163	Ramp 5-60	5
Phenol	4-Nitrophenol	980	C6H5NO3	139.027	Ramp 5-60	0
Phenol	4-Nitrophenyl_phosphate	378	C6H6NO6P	218.993	30, Ramp 5-60	1
Phenol	Catechol	289	C6H6O2	110.037	30, Ramp 5-60	3
Polyketide	Zearalenone	5281576	C18H22O5	318.147	Ramp 5-60	8
Stilbene	E-3-4-5-trihydroxy-3-glucopyranosylstilbene	5281712	C20H22O9	406.126	Ramp 5-60	5

Continued on next page.

Table A.6: Compound list for the *MassBank* dataset (continued).

group	compound	PubChem ID	molecular formula	monoisotopic mass	collision energies	annotated NLS
Sugar	2-Deoxyribose-5-phosphate	439288	C5H107P	214.024	Ramp 5-60	2
Sugar	Alpha-D-(+)-mannose-1-phosphate	439279	C6H13O9P	260.03	Ramp 5-60	2
Sugar	Alpha-D-Galactose-1-phosphate	123912	C6H13O9P	260.03	Ramp 5-60	4
Sugar	Alpha-D-Glucose-1-6-diphosphate	82400	C6H14O12P2	339.996	Ramp 5-60	6
Sugar	Alpha-D-glucose-1-phosphate	439165	C6H13O9P	260.03	Ramp 5-60	4
Sugar	D(-)-Gulono-gamma-lactone	165105	C6H10O6	178.048	Ramp 5-60	9
Sugar	D-(+)-Cellotriose	440950	C18H32O16	504.169	Ramp 5-60	22
Sugar	D-(+)-Melezitose	92817	C18H32O16	504.169	Ramp 5-60	12
Sugar	D-(+)-Raffinose	439242	C18H32O16	504.169	Ramp 5-60	9
Sugar	D-(+)-Trehalose	7427	C12H22O11	342.116	Ramp 5-60	10
Sugar	D-Arabinose-5-phosphate	230	C5H11O8P	230.019	Ramp 5-60	3
Sugar	D-Erythrose-4-phosphate	697	C4H9O7P	200.009	Ramp 5-60	3
Sugar	D-Fructose-6-phosphate	439160	C6H13O9P	260.03	Ramp 5-60	2
Sugar	D-Glucosamine-6-phosphate	439217	C6H14NO8P	259.046	Ramp 5-60	3
Sugar	D-Glucose-6-phosphate	5958	C6H13O9P	260.03	Ramp 5-60	3
Sugar	D-Mannose-6-phosphate	65127	C6H13O9P	260.03	Ramp 5-60	4
Sugar	D-Ribose-5-phosphate	439167	C5H11O8P	230.019	Ramp 5-60	3
Sugar	D-Ribulose-5-phosphate	439184	C5H11O8P	230.019	Ramp 5-60	2
Sugar	L-(+)-Rhamnose	25310	C6H12O5	164.068	Ramp 5-60	0
Sugar	Maltotriose	439586	C18H32O16	504.169	Ramp 5-60	25
Sugar	Palatinose	439559	C12H22O11	342.116	Ramp 5-60	14
Sugar	Sucrose	5988	C12H22O11	342.116	Ramp 5-60	11
Sugar alcohol	1-2-Dilauroyl-sn-Glycero-3-Phosphate	9547171	C27H53O8P	536.348	Ramp 5-60	5
Sugar alcohol	1-Lauroyl-2-Hydroxy-sn-Glycero-3-Phosphocholine	460605	C20H42NO7P	439.27	Ramp 5-60	1
Sugar alcohol	1-Myristoyl-2-Hydroxy-sn-Glycero-3-Phosphate	9547180	C17H35O7P	382.212	Ramp 5-60	3
Sugar alcohol	D-(-)-Mannitol	6251	C6H14O6	182.079	Ramp 5-60	9
Sugar alcohol	DL-Glyceraldehyde_3-phosphate	729	C3H7O6P	169.998	Ramp 5-60	3
Sugar alcohol	D-Sorbitol	5780	C6H14O6	182.079	Ramp 5-60	9
Sugar alcohol	D-Sorbitol-6-phosphate	152306	C6H15O9P	262.045	Ramp 5-60	2
Sugar alcohol	Dulcitol	11850	C6H14O6	182.079	Ramp 5-60	11
Sugar alcohol	Galactinol	439451	C12H22O11	342.116	Ramp 5-60	14
Sugar alcohol	Glycerol-2-phosphate	2526	C3H9O6P	172.014	Ramp 5-60	2
Sugar alcohol	L-Iditol	5460044	C6H14O6	182.079	Ramp 5-60	5
Sugar alcohol	Maltitol	493591	C12H24O11	344.132	Ramp 5-60	10
Sugar alcohol	Rac-Glycerol_3-phosphoate	439162	C3H9O6P	172.014	Ramp 5-60	2
	2-Hydroxyphenylacetic_acid	11970	C8H8O3	152.047	Ramp 5-60	1
	Hinokitiol	3611	C10H12O2	164.084	30, Ramp 5-60	0
	Methyl_Salicylate	4133	C8H8O3	152.047	Ramp 5-60	1

Ehrenwörtliche Erklärung

Hiermit erkläre ich

- dass mir die Promotionsordnung der Fakultät bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte oder Ergebnisse eines Dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönliche Mitteilungen und Quellen in meiner Arbeit angegeben habe,
- dass ich die Hilfe eines Promotionsberaters nicht in Anspruch genommen habe und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts haben mich folgende Personen unterstützt:

Sebastian Böcker, Markus Chimani, Kai Dührkop, Georg Pohnert, Florian Rasche, Martin Rempt, Kerstin Scheubert.

Ich habe weder die gleiche, noch eine ähnliche oder eine andere Arbeit an einer anderen Hochschule als Dissertation eingereicht.

Jena, den 25. November 2013

Franziska Hufsky